Suites définies par récurrence

1 Visualisation de suites récurrentes

Dans ce qui suit, on se donne trois fonctions d'une variable réelle :

$$f_1(x) = 1 + x - \frac{x^2}{3},$$
$$f_2(x) = \frac{-9}{49} + \frac{13x}{7} - x^2,$$
$$f_3(x) = \frac{x}{2} + \frac{3}{2x}.$$

1. Chacune de ces fonctions admet un unique point fixe strictement positif, que l'on notera respectivement x_1, x_2 et x_3. Calculer les valeurs exactes de ces points fixes.

Soit $i \in \{1, 2, 3\}$. On veut étudier les suites définies par récurrence par $u_0 = a$ et $u_{n+1} = f_i(u_n)$ pour tout $n \geq 0$, où la condition initiale a est choisie selon votre préférence dans :

- $(-\sqrt{3}, 3 + \sqrt{3})$ pour $i = 1$,
- $(3/7, 10/7)$ pour $i = 2$,
- $(0, +\infty)$ pour $i = 3$.

2. On fixe $i = 2$. Tracer u_n en fonction de n, pour $0 \leq n \leq 50$. Qu'observez-vous ?

Le *diagramme en toile d'araignée* d'une suite récurrente autonome $(u_n)_{n\geq 0}$, définie à partir d'un point de départ $u_0 = a$ et d'une fonction f, est le diagramme composé :

- du graphe de la fonction identité ;
- du graphe de la fonction f ;
- du segment $[(u_0, 0), (u_0, u_0)]$;
- pour tout n, des segments $[(u_n, u_n), (u_n, u_{n+1})]$ et $[(u_n, u_{n+1}), (u_{n+1}, u_{n+1})]$.

3. Pour chacune des fonctions f_1, f_2, f_3 et des points de départ de votre choix, dessinez les diagrammes en toile d'araignée des suites $(u_n)_{n\geq 0}$. Vous pouvez vous arrêter après 100 itérations.

2 Vitesse de convergence de suites récurrentes

On cherche à étudier la vitesse de convergence de suites définies par récurrence, et en particulier pour les trois fonctions de la partie précédente. La première étape consiste à afficher les valeurs successives de la suite $(u_n)_{n\geq 0}$. Cependant :

- on veut pouvoir calculer un grand nombre de décimales des termes de la suite. Vous allez pour cela utiliser des variables non pas de type `float`, mais de type `decimal`, ce qui permet de travailler avec une précision arbitraire.
- on veut pouvoir afficher un grand nombre de valeurs de u_n (une centaine). Afin de pouvoir les visualiser plus facilement, vous allez imprimer ces valeurs dans un fichier texte.

2.1 Variables de type Decimal

Au début de votre programme, importez le module `decimal`.

```
from decimal import *
```

Ensuite, fixez la précision, c'est-à-dire le nombre de décimales des variables de type `decimal`. Pour commencer, on utilisera une centaine de décimales :

```
getcontext().prec = 100
```

Ensuite, quand vous voudrez utiliser des variables de type `decimal`, il faudra les déclarer explicitement dans le programme. Par exemple :

```
x = Decimal(3)
```

déclare x comme une variable de type `decimal` égale à 3 (autrement dit, $x = 3.0 \ldots 0$, avec 100 zéros).

2.2 Écrire dans un fichier texte

Avant toute chose, créez un fichier `NomDeFichier.txt` dans le même répertoire que votre programme ; c'est dans ce fichier que votre programme écrira. Ensuite, pour écrire dans ce fichier, il faut, en Python :

- ouvrir ce fichier, avec les droits d'écriture ;
- écrire dans ce fichier ;
- fermer ce fichier.

Par exemple, pour écrire “Hello world!” dans `NomDeFichier.txt`, on entre le code suivant :

```
TextFile = open("NomDeFichier.txt", "w")
TextFile.write('Hello world!')
TextFile.close()
```

La commande `TextFile.write()` prend en argument une chaîne de caractères. Pour écrire dans le fichier une variable numérique x (de type `float` ou `decimal`), il faut donc écrire :

```
TextFile.write(str(x))
```

Le saut de ligne correspond à la chaîne de caractères `\n`.

2.3 Vitesse de convergence

4. Pour chacune des fonctions f_1, f_2, f_3 et des points de départ de votre choix, écrire dans un fichier texte les valeurs de u_n pour $0 \leq n \leq 100$ les unes au-dessus des autres. Vous calculerez pour commencer les valeurs de la suite avec une précision de 100 décimales.
5. Justifier à l'aide de ces calculs que la suite obtenue à l'aide de f_1 converge géométriquement, et estimer un réel $\lambda \in (0, 1)$ tel que :
$$|u_n - x_1| = O(\lambda^n).$$
6. Quelle est la nature de la convergence des suites obtenues à l'aide de f_2 et f_3 ? Rapide, géométrique ou lente ?

Finalement, il existe d'autres méthodes pour étudier la vitesse de convergence d'une suite. Dans les cas qui nous intéressent, on peut étudier $|u_n - u_{n+1}|$ au lieu de $|u_n - x_i|$; l'avantage est que u_{n+1} est facile à calculer, alors que la valeur de x_i n'est pas a priori connue.

7. Écrire un programme qui, étant donnés un entier $k \geq 0$, une condition initiale x et une fonction f, vous renvoie le plus petit entier n tel que $|u_n - u_{n+1}| \leq 10^{-k}$. Si l'entier n est au-dessus d'un seuil de votre choix, ce programme vous renverra `'Erreur : boucle trop longue.'`. Utilisez ce programme pour calculer le nombre d'itérations nécessaires pour obtenir 90 décimales avec f_1, f_2 et f_3. Comparez les résultats.
8. Écrire un programme qui, étant données une condition initiale x et une fonction f, vous affiche la suite des $|\log_{10}(|u_n - u_{n+1}|)|$ pour $0 \leq n \leq 100$. Si $u_n = u_{n+1}$ à la précision utilisée, ce programme vous renverra $|\log_{10}(\text{Précision})|$ (par exemple, 100 si vous travaillez avec 100 décimales). Tester ce programme avec f_1, f_2 et f_3.

3 Accélération de convergence

3.1 Algorithme de Newton

9. En utilisant l'algorithme de Newton, définir par récurrence une suite qui converge rapidement vers $\sqrt{13}$.

10. Utiliser cette suite pour calculer les 1000 premières décimales de $\sqrt{13}$. Vous pourrez adapter le programme obtenu à la question 7.

La fonction W de Lambert est définie comme un inverse de la fonction $x \mapsto xe^x$. Plus précisément, pour tout $z \geq -1/e$, on définit $W(z)$ comme l'unique réel supérieur ou égal à -1 tel que :

$$W(z)e^{W(z)} = z.$$

11. En utilisant l'algorithme de Newton, définir par récurrence une suite qui converge rapidement vers $W(1)$. Utiliser cette suite pour calculer les 100 premières décimales de $W(1)$.

3.2 Bassins d'attraction

Soit f une fonction de $\mathbb{R}$ dans $\mathbb{R}$, et x un point fixe de f. On rappelle que le *bassin d'attraction* de x est l'ensemble :

$$Bas_f(x) = \{y \in \mathbb{R} | \lim_{n\to+\infty} f^n(y) = x \}.$$

Comme vu en cours, si f est de classe $\mathcal{C}^1$ et x est un point fixe attractif ou super-attractif, alors $Bas_f(x)$ est un voisinage de x. Ce résultat est crucial pour la résolution numérique d'équation. Pour résoudre numériquement une équation du type

$$g(x) = \alpha,$$

une méthode courante et très efficace est de construire une fonction f telle que les solutions de l'équation précédente soient des points fixes attractifs ou super-attractifs de f. Comme $Bas_f(x_0)$ est un voisinage du point fixe x_0, si l'on sait approcher grossièrement x_0, alors on peut trouver un point de $Bas_f(x_0)$, donc une condition initiale qui convient pour cette méthode.

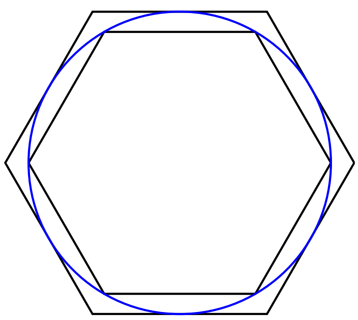
Cependant, que se passe-t-il si l'on choisit une condition initiale quelconque, potentiellement loin des points fixes de f ?

12. Prenons $f(x) = f_3(x) = \frac{x}{2} + \frac{3}{2x}$, avec $f(0) = 0$. Cette fonction a deux points fixes super-attractifs : $\pm\sqrt{3}$. Discrétisez l'intervalle $[-2, 2]$ avec une grille de 1000 points. Ensuite, écrivez un programme qui, pour chaque valeur de x dans la grille :
    - S'il existe un entier $n \leq 50$ tel que $|f^n(x) - \sqrt{3}| \leq 10^{-3}$: colorie le point en rouge.
    - S'il existe un entier $n \leq 50$ tel que $|f^n(x) + \sqrt{3}| \leq 10^{-3}$: colorie le point en bleu.
    - Sinon : colorie le point en noir.

    Affichez la figure obtenue. D'après celle-ci, quel est le bassin d'attraction de $\sqrt{3}$? De $-\sqrt{3}$?
13. Faire de même avec la fonction $f_4(x) = 2x^3/(3x^2 - 1)$. Suivant la méthode de Newton, les points fixes de f_4 sont les solutions de l'équation $x^3 - x = 0$, donc appartiennent à $\{-1, 0, 1\}$. Vous devrez donc utiliser 4 couleurs. Décrivez les bassins d'attraction que vous observez.
14. Affichez $f_4^n(x)$ en fonction de n, pour $0 \leq n \leq 50$ et les conditions initiales $x = 1/\sqrt{5} \pm 10^{-6}$. Commentez.

3.3 Méthode de Romberg-Richardson

Afin d'approcher π, une méthode consiste à inscrire dans le cercle un polygone convexe, et à circonscrire le cercle par un polygone convexe. Le périmètre du cercle est alors compris entre les périmètres des polygones inscrit et circonscrit, ce qui permet d'encadrer π.



Hexagones inscrit et circonscrit à un cercle.

Par exemple, en encadrant le cercle par des carrés, on obtient :

$$2,82 < 2\sqrt{2} \leq \pi \leq 4,$$

et l'encadrement par des hexagones fournit :

$$3 \leq \pi \leq 2\sqrt{3} < 3,47.$$

Soit u_n le demi-périmètre d'un 2^{n+2}-gone régulier inscrit dans le cercle unité, et v_n le demi-périmètre d'un 2^{n+2}-gone régulier circonscrit au cercle unité. Alors, pour tout $n \geq 0$:

$$u_n \leq \pi \leq v_n.$$

De plus, un argument de trigonométrie donne la relation de récurrence suivante :

$$\begin{cases} (u_0, v_0) & = & (2\sqrt{2}, 4), \\ (u_{n+1}, v_{n+1}) & = & \left(\sqrt{u_n v_{n+1}}, \frac{2u_n v_n}{u_n + v_n}\right) \end{cases}.$$

Autrement dit, v_{n+1} est la moyenne harmonique de (u_n, v_n), et u_{n+1} est la moyenne géométrique de (u_n, v_{n+1}).

15. Écrivez un programme qui calcule la suite (u_n, v_n), qui s'arrête dès que $(v_n - u_n) \leq 10^{-51}$, et retourne le couple (n, u_n) correspondant. Déduisez-en les 50 premières décimales de π. Combien d'itérations ont-elles été nécessaires ?

Par le même argument de trigonométrie, on sait que

$$u_n = 2^n \sin(2^{-n}\pi).$$

En particulier, en développant le sinus, $u_n = \pi - \pi^3 4^{-n}/6 + O(16^{-n})$. On pose :

$$w_n = \frac{u_{n+1} - \frac{u_n}{4}}{1 - \frac{1}{4}} = \frac{4u_{n+1} - u_n}{3}.$$

16. Modifiez le programme précédent pour qu'il calcule aussi les valeurs de w_n, s'arrête dès que $|w_{n+1} - w_n| \leq 10^{-51}$, et retourne le couple (n, w_n) correspondant. Combien d'itérations ont-elles été nécessaires ?
17. Démontrez que $w_n = \pi + O(16^{-n})$. Est-ce cohérent avec les observations précédentes ?

4 Pour aller plus loin

4.1 Suites bidimensionnelles et attracteur de Hénon

En dimension 2, les phénomènes dynamiques peuvent être beaucoup plus complexes qu'en dimension 1. On va considérer la suite définie par réccurrence :

$$\begin{cases} (u_0, v_0) & \in & [-1, 1] \times [-0.5, 0.5] \\ (u_{n+1}, v_{n+1}) & = & \left(1 - 1.4u_n^2 + v_n, 0.3u_n\right) \end{cases}.$$

18. Écrivez un programme qui, à partir des conditions initales de votre choix, vous renvoie le nuage de points correspondant aux valeurs de (u_n, v_n) pour $0 \leq n \leq 1000$.
19. Afin de voir vers quel ensemble la suite (u_n, v_n) converge, n'afficher que les valeurs de (u_n, v_n) pour $50 \leq n \leq 1000$. Cet ensemble dépend-il des conditions initiales ?

4.2 Suite logistique

Soit $\mu \in [0, 4]$. La suite logistique de paramètre μ est obtenue en prenant $f(x) = \mu x(1 - x)$, soit :

$$\begin{cases} u_0 & \in & [0, 1] \\ u_{n+1} & = & \mu u_n(1 - u_n) \end{cases}.$$

20. Démontrez que $f([0, 1]) \subset [0, 1]$.
21. Soit $\mu \in [0, 3]$. Tracez le diagramme en araignée correspondant à la suite $(u_n)_{n \geq 0}$ pour $0 \leq n \leq 50$ et une condition initiale de votre choix. Que pouvez-vous conjecturer sur le comportement de la suite $(u_n)_{n \geq 0}$?
22. Faitez de même avec $\mu = 4$.
23. On garde la valeur $\mu = 4$. Prenez une condition initiale au hasard, et tracez l'histogramme des valeurs de (u_n) pour $0 \leq n \leq 10000$, dont les classes sont les intervalles $[k/100, (k + 1)/100]$ pour $0 \leq k < 100$. Cet histogramme dépend-t-il de la condition initiale ? Que pouvez-vous conjecturer sur le comportement de la suite (u_n) ?
24. Démontrez votre conjecture de la question 21.