Ce problème porte sur l'étude de la suite logistique de Verhulst. Une des composantes principales de ce problème est le tracé de la suite pour différents taux de croissance. Il est attendu que chaque graphique possède un titre, des axes précisément spécifiés et que les courbes figurant sur les graphiques soient légendées. Pour les tracés, la bibliothèque `matplotlib.pyplot` sera utilisée et on pourra aussi, si besoin, utiliser les fonctionnalités disponible dans la bibliothèque `numpy`. Chaque fonction possèdera un en-tête précisant à minima la fonctionnalité précise de la fonction.

1 Présentation de la suite logistique de Verhulst

En dynamique des populations, le modèle de Verhulst est un modèle de croissance proposé par Pierre François Verhulst vers 1840. Verhulst a proposé ce modèle en réponse au modèle de Malthus qui proposait un taux d'accroissement constant sans frein conduisant à une croissance exponentielle de la population.

Le modèle de Verhulst imagine que le taux de natalité et le taux de mortalité sont des fonctions affines respectivement décroissante et croissante de la taille de la population. Autrement dit, plus la taille de la population augmente, plus son taux de natalité diminue et son taux de mortalité augmente. Verhulst pose d'autre part que, lorsque les populations sont de petites tailles, elles ont tendance à croître.

Le même modèle est utilisable pour des réactions autocatalytiques, dans lesquelles l'augmentation des individus touchés est proportionnelle à la fois au nombre d'individus déjà touchés et au nombre d'individus qui peuvent encore être touchés.

Ce modèle conduit, en temps continu, à une fonction logistique et en temps discret à une suite logistique dont la particularité est d'être, dans certaines circonstances, chaotique.

Dans ce problème, on s'intéresse spécifiquement à la suite logistique :

$$u_{n+1} = ru_n(1 - u_n)$$

Sous cette forme, u_n représente une fraction d'une population. Par exemple la fraction de cellules tumorales dans un échantillon biologique. r est appelé le taux de croissance de la suite.

Le problème se propose de réaliser une étude du comportement de la suite pour différentes valeurs de r. Sauf indication contraire, on prendra $u_0 = 0.05$. Chaque itération correspond à une durée de 1 jour

2 Comportement de la suite logistique pour $r \in [0, 1]$

1. Montrer que la suite converge et que sa limite est 0.
2. Définir en langage Python une fonction `croissance(u0,t,r)` avec t un nombre entier de jours, et qui renvoie la liste des dates ainsi que la liste des valeurs successives prises par u_n après chaque itération depuis l'instant initial et jusqu'à t.
3. Quelle est la complexité temporelle de cette fonction ? Justifier.
4. Rédiger un ensemble d'instructions permettant de tracer sur une même courbe en figure 0, le comportement de la suite pour une durée maximale de 100 jours et pour différents taux de croissance $r \in [0, 1]$.
5. Les courbes obtenues ont-elles bien le comportement souhaité ?
6. Afin de vérifier visuellement la limite de la suite pour $r = 1$ quelle modification doit-t-on opérer ?.
7. Tracer en figure 1 le comportement de u_n de 0 à 1000 jours. L'allure de la courbe est-elle en accord avec la prédiction théorique ?

3 Comportement de la suite logistique pour $r \in]1, 2]$

8. Tracer de même dans une nouvelle fenêtre (figure 2) l'évolution de la suite logistique pour différents taux de croissance compris en 1 (exclus) et 2 (inclus).
9. La suite est-elle toujours convergente ?
10. Déterminer alors la limite de la suite pour $r \in]1, 2]$. Vérifier numériquement sa valeur sur les taux de croissances testés.

4 Comportement de la suite logistique pour $r \in]2, 3]$

11. Tracer de même dans une nouvelle fenêtre (figure 3) l'évolution de la suite logistique pour différents taux de croissance compris en 2 (exclus) et 3 (inclus).
12. La suite est-elle toujours convergente ?
13. Observer (en figure 4) sur une durée de 1000 jours le comportement de la suite pour un taux de croissance égal à 3.
14. Vérifier graphiquement que l'expression de la limite trouvée précédemment est toujours valable.

5 Comportement de la suite logistique pour $r \in]3, 4]$

15. Tracer dans une dans une nouvelle fenêtre (figure 5) l'évolution de la suite logistique pour un taux de croissance égal à 3,1 sur une durée de 1000 jours.
16. Quelle différence majeure observe-t-on avec le cas où $r = 3$
17. Tracer dans une nouvelle fenêtre (fenêtre 6) et sur un même graphe l'évolution de la suite logistique durant 100 jours, pour un taux de croissance de 3,1 et pour deux conditions initiales très proches : $u_0 = 0.05$ et $u'_0 = 0.05 + 1 \times 10^{-6}$
18. Cette légère différence dans les conditions initiales a-t-elle un impact significatif sur l'évolution ultérieur de la suite ?
19. Reprendre les deux questions précédentes pour $r = 3,8$. (graphe affiché en figure 7).
20. Comment qualifier l'évolution de la population dans le cas où $r = 3.8$?

6 Diagramme de Feigenbaum

Le comportement de la suite logistique pour différents taux de croissance peut être résumé sur le diagramme de Feigenbaum : On reporte en ordonnées les valeurs que la suite logistique peut prendre une fois le régime stationnaire atteint. Idéalement on prend la/les valeur(s) que peut prendre la suite pour n infini.

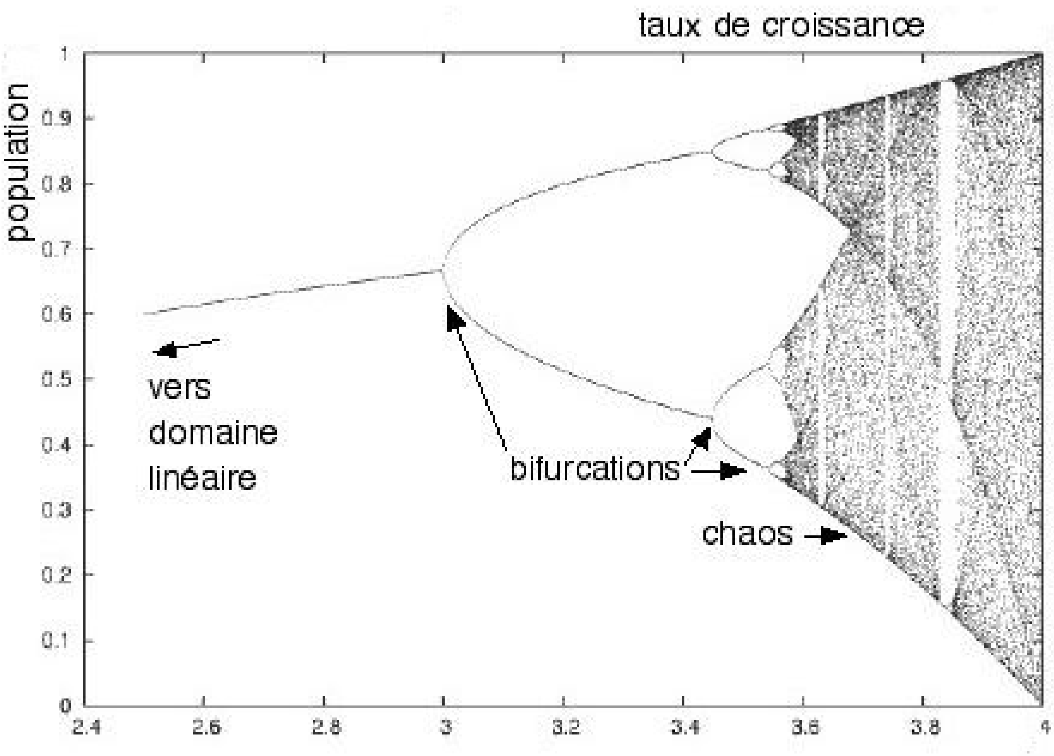

Figure 1 – Diagramme de Feigenbaum

21. Expliquer l'allure de ce graphe.

22. On considère que le régime stationnaire est atteint après moins de 1000 itérations.
Définir une fonction `croissance1000(u0,r)` qui renvoie la liste $[u_{1000}, u_{1001}, \cdots, u_{1010}]$
23. Quelle est la complexité de la fonction `croissance1000(u0,r)` ?
24. Définir une fonction `chemins(u0,rmin,rmax,N)` qui renvoie, une liste contenant, pour N valeurs de r comprises entre $rmin$ et $rmax$, la liste des 10 valeurs obtenues en appliquant la fonction `croissance1000(u0,r)`
25. Quelle est la complexité de la fonction `chemins(u0,rmin,rmax,N)` ?
26. Définir une fonction `feigenbaum(u0,rmin,rmax,N)` qui utilise la fonction `chemins` et qui trace le diagramme de Feigenbaum.
27. Quelle est la complexité de la fonction `feigenbaum(u0,rmin,rmax,N)` ?
28. Tracer le diagramme de Feigenbaum.