Mini-projet 2 : un exemple de système chaotique

Le lien vers la page du problème se trouve facilement ci-dessous :

`phys-mod.github.io/source/notebooks/suites-relations-recurrence/mini-projet-logistique.html`

Le problème décrit concerne la croissance d'une population de lapins, qui est un nombre entier positif et discret. Le problème de représentation du nombre de lapins passe d'un cas discret à un cas continu lorsque nous effectuons la transformation $x \equiv n/n_{\max}$. À partir de ce moment, nous ne devons plus nous préoccuper de n ni de $n_{\max}$. Vous devez commencer avec une valeur de `x0 = 0.00001` (ou `x0 = 1e-5`). Ce sera le premier terme de la suite. Le terme suivant de la suite est obtenu par

$$x_{i+1} = f(x_i)$$

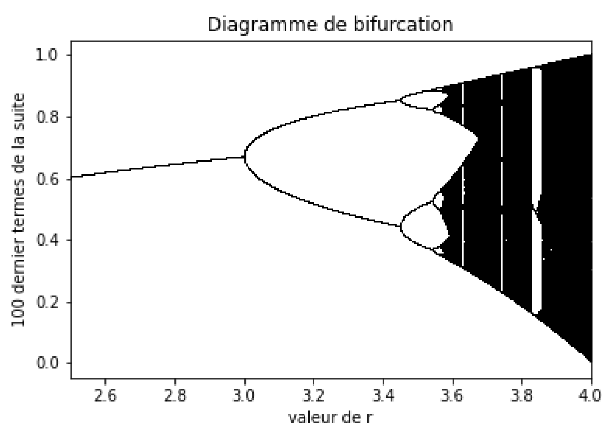
Où $f(x) = r(1-x)x$. Pour chaque valeur de r, calculez les 1000 premiers termes de la suite.

Pour vérifier que votre programme fonctionne correctement jusqu'à ce stade, tracez différents graphiques pour cette séquence avec différentes valeurs de r. Par exemple : `r = 2.5, 3.2, 3.5, 4.0`. Vous devez constater que la série converge non seulement vers une valeur fixe comme dans `r = 2.5`, mais aussi vers deux dans `r = 3.2`, puis vers quatre dans `r = 3.0` et enfin, lorsque `r = 4.0`, la valeur des termes de la série est complètement chaotique.

Astuce : au lieu d'utiliser `plt.plot(...)`, utilisez `plt.scatter(...)`. Vous verrez ainsi plus clairement le comportement chaotique de la série pour les valeurs de r proches de 4.

Trouvez un moyen de stocker les 100 derniers termes de la série pour chaque valeur de r (vous pouvez sélectionner les `N` derniers éléments de votre tableau `liste` en écrivant `liste[-N:]`).

Vous devez maintenant afficher un graphique des 100 dernières valeurs de la série pour chaque valeur de r. Tracez un graphique des valeurs en fonction de r. Vous devriez obtenir quelque chose qui ressemble à la figure ci-dessous.



Pour l'exposant de Lyapunov, nous devons le calculer de la manière suivante

$$\lambda = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} \ln \left| \frac{\mathrm{d}f(x_i)}{\mathrm{d}x} \right|$$

C'est la valeur de l'exposant pour la suite pour une valeur fixe de r. Il faut maintenant montrer comment la valeur de λ change en fonction de r.

Il n'est pas possible de calculer numériquement la limite avec une somme infinie à l'aide de Python. Dans ce cas, vous pouvez utiliser les 100 derniers termes pour calculer la valeur de λ.

Astuce : définissez une fonction qui est la dérivée de $f(x)$ à utiliser pour calculer λ. Pour vous faciliter la vie, calculez la dérivée analytiquement. **Attention :** vous devez calculer le **logarithme naturel** du module de la dérivée de la fonction en x_i (`np.log()` de la bibliothèque NumPy).

Vous devriez trouver un graphique similaire à celui ci-dessous