

~\Documents\Skola dokue\Master Year 1\DynamicStuff master 1\HW1\1.1.py

```
1 #%% Import libraries
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from scipy.optimize import fsolve
5 from scipy.optimize import root_scalar
6
7 #%% Plot bifurcation
8 # Define the dimensionless equation
9 def dxdt(x, r):
10     return 1/5 + 7/(10 * (1 + np.exp(80 * (1 - x)))) - r * x**4
11
12 def stability(x, r):
13     h = 1e-5
14     return (dxdt(x + h, r) - dxdt(x, r)) / h
15
16 r_values = np.linspace(0.01, 1.5, 600)
17
18 fixed_points = []
19 stability_info = []
20
21 for r in r_values:
22     guesses = np.linspace(0, 1.5, 10)
23     solutions = []
24     for guess in guesses:
25         sol, _, ier, _ = fsolve(dxdt, guess, args=(r,), full_output=True)
26         if ier == 1:
27             sol = np.round(sol, 6)
28             if sol not in solutions:
29                 solutions.append(sol)
30
31     stability_status = [stability(x, r) for x in solutions]
32     fixed_points.append(solutions)
33
34     stability_info.append(stability_status)
35
36 plt.figure(figsize=(10, 6))
37 for i, r in enumerate(r_values):
38     for x, stab in zip(fixed_points[i], stability_info[i]):
39         if stab < 0: # Stable
40             plt.plot(r, x, 'bo', markersize=2)
41         else: # Unstable
42             plt.plot(r, x, 'ro', markersize=2)
43             print(f'rvalue: {r}, xvalue: {x}')
44
45 plt.title("Bifurcation Diagram")
46 plt.xlabel("r")
47 plt.ylabel("x*")
48 plt.axhline(0, color="black", linewidth=0.5, linestyle="--")
49 plt.grid()
50 plt.show()
```

