

## 2.3\2.3.py

```
1 #%% Import libraries
2 import numpy as np
3 import matplotlib.pyplot as plt
4 #%% a)
5 def jacobian(x, alpha):
6     J = np.array([[0, 1],
7                  [-np.cos(x), -alpha]])
8     return J
9
10 def classify_fixed_points(alpha):
11     fixed_points = [0, np.pi]
12     for x_fp in fixed_points:
13         J = jacobian(x_fp, alpha)
14         eigenvalues, _ = np.linalg.eig(J)
15
16         if np.all(np.real(eigenvalues) < 0) and eigenvalues[0] != eigenvalues[1]:
17             stability = "Stable (Sink)"
18         elif np.all(np.real(eigenvalues)< 0):
19             stability = "Degenerate"
20         elif np.all(np.real(eigenvalues) > 0):
21             stability = "Unstable (Source)"
22         else:
23             stability = "Saddle Point"
24
25         print(f"Fixed point x = {x_fp:.2f}, Eigenvalues: {eigenvalues}, Stability: {stability}")
26
27 alphas = [0, 0.5, 1, 2]
28
29 for alpha in alphas:
30     print(f"\nAlpha = {alpha}")
31     classify_fixed_points(alpha)
32
33 #%% b)
34
35 import numpy as np
36 import matplotlib.pyplot as plt
37
38 def pendulum_system_vector_field(x, y, alpha):
39     dxdt = y
40     dydt = -np.sin(x) - alpha * y
41     return dxdt, dydt
42
43 def plot_phase_portrait_stream(ax, alpha, fixed_point, title):
44     ax.set_title(title)
45     ax.set_xlabel("x")
46     ax.set_ylabel("y")
47
48     x_values = np.linspace(fixed_point - np.pi / 2, fixed_point + np.pi / 2, 100)
49     y_values = np.linspace(-2, 2, 100)
50     X, Y = np.meshgrid(x_values, y_values)
51
```

```
52     U, V = pendulum_system_vector_field(X, Y, alpha)
53
54     ax.streamplot(X, Y, U, V, color='b', density=1.5, linewidth=1)
55
56     ax.plot(fixed_point, 0, 'ro', markersize=10)
57     ax.grid(True)
58     ax.set_xlim([fixed_point - np.pi / 2, fixed_point + np.pi / 2])
59     ax.set_ylim([-2, 2])
60
61 fixed_points = [0, np.pi]
62
63 alphas = [0, 0.5, 1, 2]
64
65 for alpha in alphas:
66     fig, axs = plt.subplots(1, 2, figsize=(12, 6))
67     fig.suptitle(f"Phase Portraits for  $\alpha = \{alpha\}$ ", fontsize=16)
68
69     for j, fixed_point in enumerate(fixed_points):
70         title = f"x = {fixed_point:.2f}"
71         plot_phase_portrait_stream(axs[j], alpha, fixed_point, title)
72
73     plt.tight_layout()
74     plt.show()
```