# Homework Assignment 2.3

Max Green TIF155

December 1, 2024

## Exercise a

The system of differential equations is given by:

$$\frac{dx}{d\tau} = y,$$

$$\frac{dy}{d\tau} = -\sin(x) - \alpha y.$$

The partial derivatives are:

$$\frac{\partial f_1}{\partial x} = 0, \quad \frac{\partial f_1}{\partial y} = 1,$$

$$\frac{\partial f_2}{\partial x} = -\cos(x), \quad \frac{\partial f_2}{\partial y} = -\alpha.$$

Thus, the Jacobian matrix is:

$$J(x,y) = \begin{bmatrix} 0 & 1 \\ -\cos(x) & -\alpha \end{bmatrix}.$$

$$\det \begin{bmatrix} -\lambda & 1 \\ -\cos(x) & -\alpha-\lambda \end{bmatrix} = 0.$$

Expanding the determinant:

$$(-\lambda)(-\alpha-\lambda) - (1)(-\cos(x)) = 0,$$

$$\lambda^2 + \alpha\lambda + \cos(x) = 0.$$

The eigenvalues are:

$$\lambda_{1,2} = \frac{-\alpha \pm \sqrt{\alpha^2 - 4\cos(x)}}{2}.$$

| $\alpha$ values | Fixed point | Classification | Explanation |
|---|---|---|---|
| $< 2$ | (0,0) | (Stable) Spiral | Both eigenvalues imaginary ($Re(\lambda_{1,2}) > 0$) |
| 2 | (0,0) | Degenerate | $\lambda_1 = \lambda_2$ |
| $> 2$ | (0,0) | Stable | $\mathrm{Re}(\lambda_{1,2}) < 0$ |
| $\geq 0$ | ($\pi$,0) | Saddle point | One $\lambda > 0$ and one $\lambda < 0$ |

Table 1: Table of $\alpha$ values and the classication of fixed points.

# Exercise b

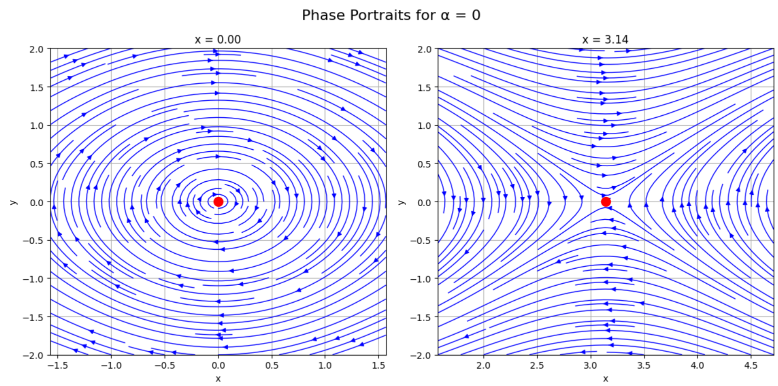


Figure 1: Plot of the representative trajectories with $\alpha = 0$.

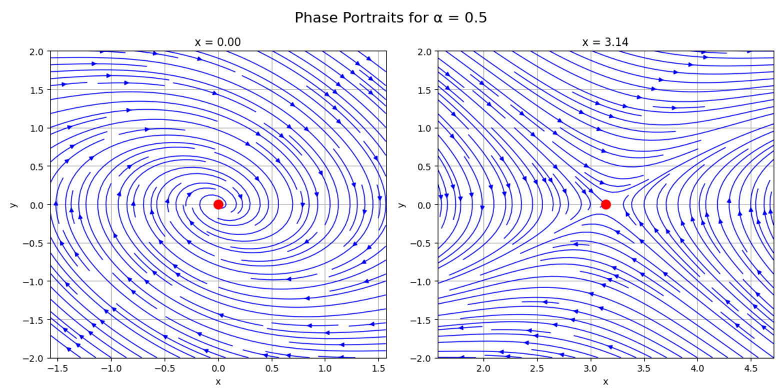


Figure 2: Plot of the representative trajectories with $\alpha = 0.5$.

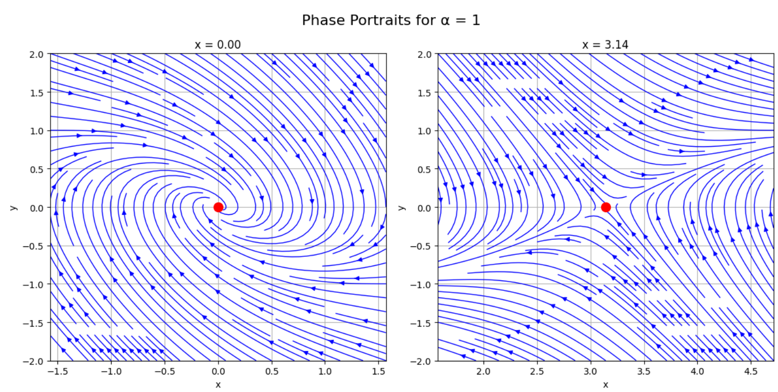


Figure 3: Plot of the representative trajectories with $\alpha = 1$.

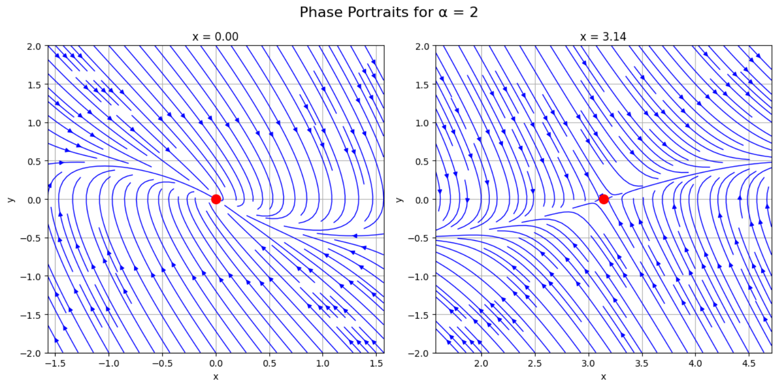


Figure 4: Plot of the representative trajectories with $\alpha = 2$.

**2.3\2.3.py**

```
#%% Import libraries
import numpy as np
import matplotlib.pyplot as plt
#%% a)
def jacobian(x, alpha):
    J = np.array([[0, 1],
                  [-np.cos(x), -alpha]])
    return J

def classify_fixed_points(alpha):
    fixed_points = [0, np.pi]
    for x_fp in fixed_points:
        J = jacobian(x_fp, alpha)
        eigenvalues, _ = np.linalg.eig(J)

        if np.all(np.real(eigenvalues) < 0) and eigenvalues[0] != eigenvalues[1]:
            stability = "Stable (Sink)"
        elif np.all(np.real(eigenvalues)< 0):
            stability = "Degenerate"
        elif np.all(np.real(eigenvalues) > 0):
            stability = "Unstable (Source)"
        else:
            stability = "Saddle Point"

        print(f"Fixed point x = {x_fp:.2f}, Eigenvalues: {eigenvalues}, Stability: {stability}")

alphas = [0, 0.5, 1, 2]

for alpha in alphas:
    print(f"\nAlpha = {alpha}")
    classify_fixed_points(alpha)

#%% b)

import numpy as np
import matplotlib.pyplot as plt

def pendulum_system_vector_field(x, y, alpha):
    dxdt = y
    dydt = -np.sin(x) - alpha * y
    return dxdt, dydt

def plot_phase_portrait_stream(ax, alpha, fixed_point, title):
    ax.set_title(title)
    ax.set_xlabel("x")
    ax.set_ylabel("y")

    x_values = np.linspace(fixed_point - np.pi / 2, fixed_point + np.pi / 2, 100)
    y_values = np.linspace(-2, 2, 100)
    X, Y = np.meshgrid(x_values, y_values)

```

```python
    U, V = pendulum_system_vector_field(X, Y, alpha)

    ax.streamplot(X, Y, U, V, color='b', density=1.5, linewidth=1)

    ax.plot(fixed_point, 0, 'ro', markersize=10)
    ax.grid(True)
    ax.set_xlim([fixed_point - np.pi / 2, fixed_point + np.pi / 2])
    ax.set_ylim([-2, 2])

fixed_points = [0, np.pi]

alphas = [0, 0.5, 1, 2]

for alpha in alphas:
    fig, axs = plt.subplots(1, 2, figsize=(12, 6))
    fig.suptitle(f"Phase Portraits for α = {alpha}", fontsize=16)

    for j, fixed_point in enumerate(fixed_points):
        title = f"x = {fixed_point:.2f}"
        plot_phase_portrait_stream(axs[j], alpha, fixed_point, title)

    plt.tight_layout()
    plt.show()
```