Homework Assignment 2.1

Max Green TIF155

December 1, 2024

Exercise a

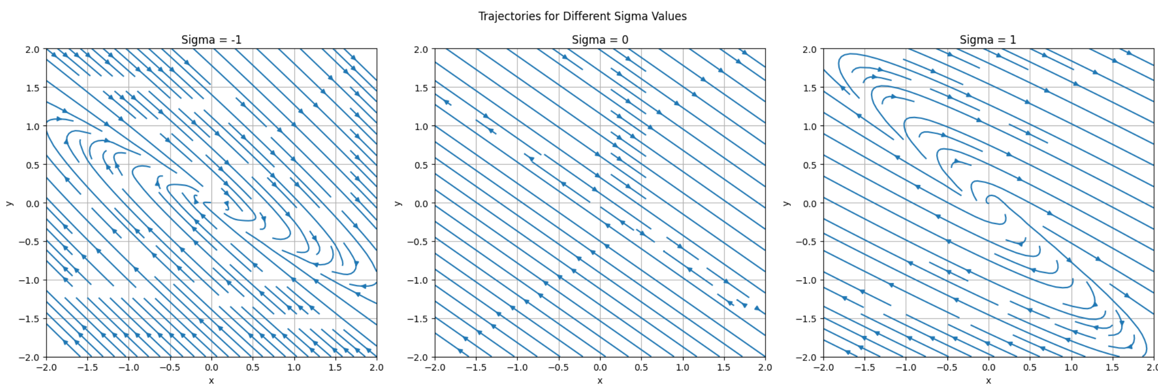

Figure 1: Plot of the representative trajectories. The fixed point at the origin is, in order, stable, a saddle point and unstable.

Exercise b

The dynamical system is represented by the matrix:

$$\mathbf{A}_\sigma = \begin{pmatrix} \sigma + 3 & 4 \\ -\frac{9}{4} & \sigma - 3 \end{pmatrix}$$

The eigenvalues of $\mathbf{A}_\sigma$ are determined by solving the characteristic equation:

$$\det(\mathbf{A}_\sigma - \lambda \mathbf{I}) = 0$$

$$\mathbf{A}_\sigma - \lambda \mathbf{I} = \begin{pmatrix} \sigma + 3 - \lambda & 4 \\ -\frac{9}{4} & \sigma - 3 - \lambda \end{pmatrix}$$

$$\det(\mathbf{A}_\sigma - \lambda \mathbf{I}) = (\sigma + 3 - \lambda)(\sigma - 3 - \lambda) + 9$$

$$\det(\mathbf{A}_\sigma - \lambda \mathbf{I}) = \lambda^2 - 2\lambda\sigma + \sigma^2$$

Setting the determinant to zero, we solve:

$$\lambda^2 - 2\lambda\sigma + \sigma^2 = 0$$

$$(\lambda - \sigma)^2 = 0$$

Thus, the eigenvalues are:

$$\lambda_1 = \lambda_2 = \sigma$$

Exercise c

The eigenvector $\mathbf{v}$ satisfies:

$$(\mathbf{A}_\sigma - \sigma\mathbf{I})\mathbf{v} = \mathbf{0}$$

Substituting $\mathbf{A}_\sigma - \sigma\mathbf{I}$, we have:

$$\mathbf{A}_\sigma - \sigma\mathbf{I} = \begin{pmatrix} 3 & 4 \\ -\frac{9}{4} & -3 \end{pmatrix}$$

$$\begin{pmatrix} 3 & 4 \\ -\frac{9}{4} & -3 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$3v_1 + 4v_2 = 0, \quad -\frac{9}{4}v_1 - 3v_2 = 0$$

$$v_2 = -\frac{3}{4}v_1$$

$$\mathbf{v} = \begin{pmatrix} v_1 \\ -\frac{3}{4}v_1 \end{pmatrix}$$

$$\|\mathbf{v}\| = \sqrt{v_1^2 + \left(-\frac{3}{4}v_1\right)^2} = \sqrt{v_1^2 + \frac{9}{16}v_1^2} = \sqrt{\frac{25}{16}v_1^2} = \frac{5}{4}|v_1|$$

$$\mathbf{v}_\sigma = \frac{1}{\frac{5}{4}|v_1|} \begin{pmatrix} v_1 \\ -\frac{3}{4}v_1 \end{pmatrix} = \begin{pmatrix} \frac{4}{5} \\ -\frac{3}{5} \end{pmatrix}$$

Exercise d

The inverse of $\mathbf{A}_\sigma$ is given by:

$$\mathbf{A}_\sigma^{-1} = \frac{1}{\det(\mathbf{A}_\sigma)} \begin{pmatrix} \sigma - 3 & -4 \\ \frac{9}{4} & \sigma + 3 \end{pmatrix}$$

Substitute $\det(\mathbf{A}_\sigma) = \sigma^2$:

$$\mathbf{A}_\sigma^{-1} = \frac{1}{\sigma^2} \begin{pmatrix} \sigma - 3 & -4 \\ \frac{9}{4} & \sigma + 3 \end{pmatrix}$$

Exercise e

This result is valid for $\sigma \neq 0$, since the determinant is zero when $\sigma = 0$

Exersice f

See exersice c.

Exersice g

To find the fixed points of the system, we solve the equations where $\dot{x}$ and $\dot{y}$ are equal to zero.

$$\dot{x} = 3x + 4y - \frac{x^3}{100} = 0$$

$$\dot{y} = -\frac{9}{4}x - 3y = 0$$

$$y = -\frac{3}{4}x. \tag{3}$$

$$3x + 4\left(-\frac{3}{4}x\right) - \frac{x^3}{100} = 0$$

$$3x - 3x - \frac{x^3}{100} = 0$$

$$-\frac{x^3}{100} = 0 \quad x^3 = 0$$

$$y = -\frac{3}{4}(0) = 0$$

The only fixed point is:

$$[x^*, y^*] = [0, 0]$$

2.1\2.1.py

```python
#%% Import libraries
import numpy as np
import matplotlib.pyplot as plt

def derivative(x, y, A_sigma):
    return A_sigma @ np.array([x, y])

def classify_stability(eigenvalues):
    real_parts = np.real(eigenvalues)
    imag_parts = np.imag(eigenvalues)

    if all(real_parts < 0):
        return "Stable Node"
    elif all(real_parts > 0):
        return "Unstable Node"
    elif real_parts[0] * real_parts[1] < 0:
        return "Saddle Point"
    elif imag_parts[0] != 0:
        if real_parts[0] == 0:
            return "Center"
        elif real_parts[0] < 0:
            return "Stable Focus"
        else:
            return "Unstable Focus"
    else:
        return "Other"

#%% Run and plot
sigmaList = [-1, 0, 1]
x = np.linspace(-2, 2, 20)
y = np.linspace(-2, 2, 20)
X, Y = np.meshgrid(x, y)

# Create subplots: 1 row, 3 columns
fig, axes = plt.subplots(1, 3, figsize=(18, 6))
fig.suptitle("Trajectories for Different Sigma Values")

for idx, sigma in enumerate(sigmaList):
    A_sigma = np.array([[sigma + 3, 4], [-9/4, sigma - 3]])
    dX, dY = np.zeros(X.shape), np.zeros(Y.shape)

    # Calculate derivatives
    for i in range(X.shape[0]):
        for j in range(X.shape[1]):
            dx, dy = derivative(X[i, j], Y[i, j], A_sigma)
            magnitude = np.sqrt(dx**2 + dy**2)
            if magnitude > 0:
                dX[i, j], dY[i, j] = dx / magnitude, dy / magnitude

    # Plot vector field in the corresponding subplot
    ax = axes[idx]
```

```
    ax.streamplot(X, Y, dX, dY)
    ax.set_title(f"Sigma = {sigma}")
    ax.set_xlabel("x")
    ax.set_ylabel("y")
    ax.grid()

    # Stability analysis
    eigenvalues, _ = np.linalg.eig(A_sigma)
    stability = classify_stability(eigenvalues)
    print(f"Sigma = {sigma}: {stability}")

# Adjust layout
plt.tight_layout()
plt.show()
```