

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

про виконання лабораторної роботи №1

по дисципліні «Нелінійний аналіз»

на тему

«Положення рівноваги та зображення фазових портретів на площині»

Виконала:

студент групи КМ-81

Верзун П.В.

Перевірив:

Балюнов О.О.

Київ — 2021

## **Варіант №2**

**Завдання 1:** дослідити положення рівноваги даної системи диференціальних рівнянь та зобразити її фазові траєкторії на площині  $XOY$

$$x' = x - 3y$$

$$y' = 7x - 9y$$

Для початку, необхідно знайти особливу точку, склавши систему рівнянь та знайти її розв'язки

$$x' = x - 3y \Rightarrow x = 0$$

$$y' = 7x - 9y \Rightarrow y = 0$$

Тож далі досліджуватимемо поведінку системи в околі особливої точки  $(0;0)$ .

Класифікація положень рівноваги визначається власними значеннями  $\lambda_1, \lambda_2$  матриці  $A$

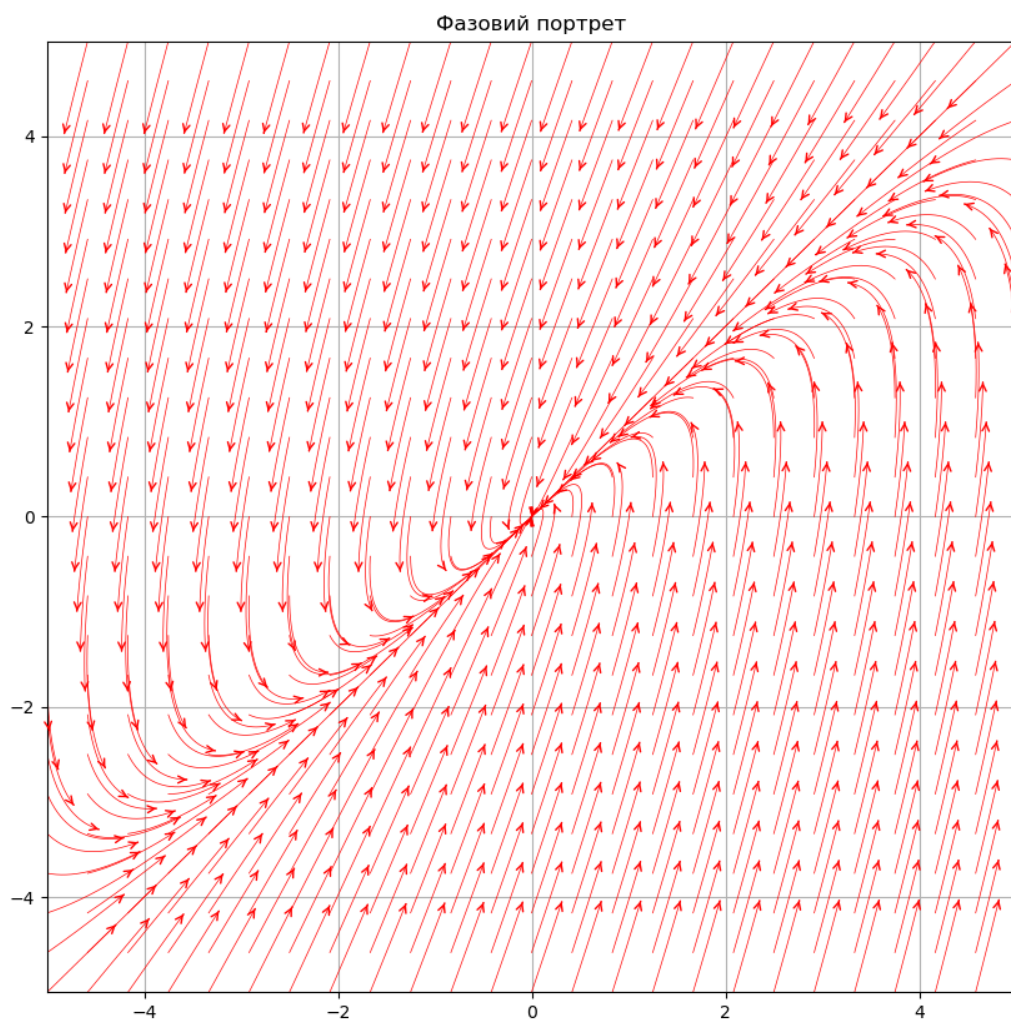
$$A = \begin{pmatrix} 1 & -3 \\ 7 & -9 \end{pmatrix}$$

$$\lambda^2 + 8\lambda + 12 = 0$$

$$\lambda_1 = -2$$

$$\lambda_2 = -6$$

таке положення називають *стійкий вузол*.



## ВИСНОВКИ

При виконанні даної лабораторної роботи було розроблено програмне забезпечення, яке дозволяє будувати фазові криві. В процесі розроблення ПЗ були використані сучасні бібліотеки мови python, що дозволили створювати візуалізацію для фазового портрету.

## Додаток А

### Лістинг програми

```
import numpy as np
import matplotlib.pyplot as plt
P = lambda x, y: x-2*y
Q = lambda x, y: 2*x+y
def phase_portrait(P, Q, N=10000,
                  x0=0, y0=0,
                  alpha_x=10**-3, alpha_y=10**-3, **kwargs):
    xi = [x0]
    yi = [y0]
    for i in range(N):
        Vx = P(xi[-1], yi[-1])
        Vy = Q(xi[-1], yi[-1])
        Vx_ = Vx / np.sqrt(Vx**2 + Vy**2)
        Vy_ = Vy / np.sqrt(Vx**2 + Vy**2)
        xi.append(xi[-1] + alpha_x * Vx_)
        yi.append(yi[-1] + alpha_y * Vy_)
    return xi, yi
x0 = np.random.uniform(-5, 5)
y0 = np.random.uniform(-5, 5)
xline, yline = phase_portrait(P, Q, x0=x0, y0=y0)
plt.grid()
plt.plot(xline, yline, label='Траєкторія')
plt.scatter(xline[:1], yline[:1], c='r', label='Початкова точка')
plt.annotate("", xy=(xline[-1], yline[-1]),
             xytext=(xline[-2], yline[-2]),
             arrowprops=dict(arrowstyle="->"))
plt.legend()
plt.title('Траєкторія, що починається в довільній точці')
def plot_phase_portrait(P, Q, evolution_func=phase_portrait,
                       num_iter=1000,
                       xmin=-5, xmax=5,
                       ymin=-5, ymax=5,
                       num_x=25, num_y=25,
                       alpha_x=10**-3, alpha_y=10**-3,
                       generator_x=lambda x:1, generator_y=lambda x:1,
                       linewidth=0.5, c='r', show_arrow=True):
    x = np.linspace(xmin, xmax, num_x)
    y = np.linspace(ymin, ymax, num_y)
    xv, yv = np.meshgrid(x, y)
    xi, yi = evolution_func(P, Q, N=num_iter,
                           x0=xv.flatten(), y0=yv.flatten(),
                           alpha_x=alpha_x, alpha_y=alpha_y,
                           generator_x=generator_x, generator_y=generator_y)
    xi, yi = np.array(xi), np.array(yi)
    plt.figure(figsize=(10, 10))
    plt.title('Фазовий портрет')
    plt.plot(xi, yi, c=c, linewidth=linewidth)
    if show_arrow:
        for i in range(xi.shape[1]):
            plt.annotate("", xy=(xi[-1, i], yi[-1, i]),
                        xytext=(xi[-2, i], yi[-2, i]),
                        arrowprops=dict(arrowstyle="->", color=c))
    plt.xlim(xmin, xmax)
```

```
plt.ylim(ymin, ymax)
plt.grid()

plot_phase_portrait(P, Q)
plot_phase_portrait(P, Q,
                    xmin=0.5-0.11,
                    xmax=0.5+0.1,
                    ymin=1-0.1,
                    ymax=1+0.1,
                    alpha_x=10**-5,
                    alpha_y=10**-5)
plot_phase_portrait(P, Q,
                    xmin=0.5-0.11,
                    xmax=0.5+0.1,
                    ymin=-1-0.1,
                    ymax=-1+0.1,
                    alpha_x=10**-5,
                    alpha_y=10**-5)

plt.show()
```