Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики

Лабораторна робота
з курсу
«Управління динамічними системами»
на тему

«Аналітичне розв'язування диференціальних рівнянь за допомогою комп'ютерних пакетів програм. Аналітичне конструювання регуляторів. Побудова фазових портретів.»

Виконав:
студент групи ІПС-21
факультету комп'ютерних наук
та кібернетики
Ольховатий Ігор

Зміст

Завдання: згідно з варіантом

- \- Дослідити на стійкість задану систему. Визначити вигляд точки спокою.

  Намалювати фазовий портрет. (Все аналітично в зошиті).

- \- Розв'язати задачу модального керування (*непарні варіанти*);

  або задачу аналітичного конструювання регуляторів (*парні варіанти*), обравши одне керування з знайдених можливих. Визначити вигляд отриманої точки спокою. Намалювати фазовий портрет. (Все аналітично в зошиті).

- \- Зобразити фазові портрети особливих точок розімкненої системи та побудованої замкненої системи за допомогою програмних пакетів (бажано **Sage**). Траєкторії, сепаратриси, ізокліни (де треба) – різний колір та товщина.

Варіант №1

$$\begin{cases} \dot{x} = 3x \\ \dot{y} = 2x + y \end{cases}$$

$$\lambda_1 = -1, \lambda_2 = -3$$

$$b = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Варіант 1

$$\begin{cases} \dot{x} = 3x \\ \dot{y} = 2x + y \end{cases} \qquad \begin{matrix} \lambda_1 = -1 \\ \lambda_2 = -3 \end{matrix} \qquad b = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Розв'язання:

У нашому випадку $A = \begin{pmatrix} 3 & 0 \\ 2 & 1 \end{pmatrix}$. Тоді

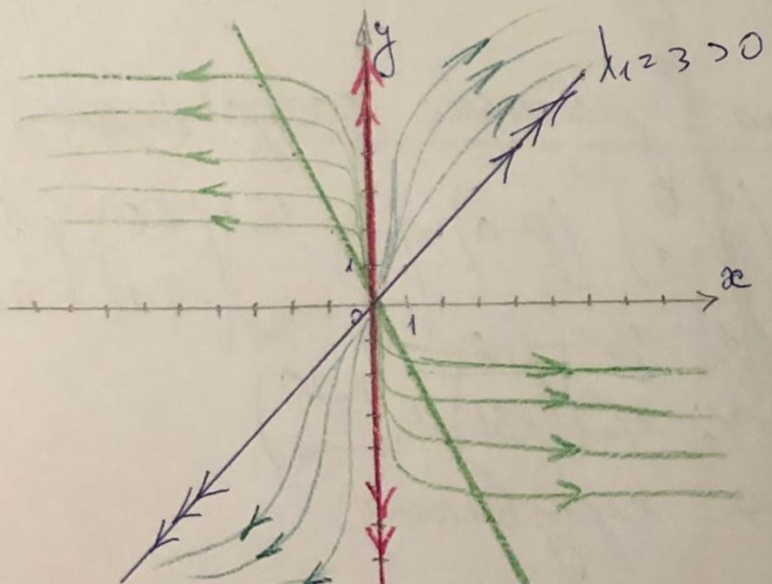
$$\det|A - \lambda E| = \begin{vmatrix} 3-\lambda & 0 \\ 2 & 1-\lambda \end{vmatrix} = 0 \Rightarrow x^2 - 4x + 3 = 0$$

$x_1 = 1$, $x_2 = 3$, тоді

$\lambda_1 = 3$, $\lambda_2 = 1$ — дійсні, різні, > 0 і одного знаку, тобто задана система НЕ стійка, а вигляд точки спокою — не стійкий вузол.

$\lambda_1 = 3$ $\begin{pmatrix} 0 & 0 \\ 2 & -2 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = 0$ $\quad \alpha = 1$, $\beta = 1$. $\quad y = x$

$\lambda_2 = 1$ $\begin{pmatrix} 2 & 0 \\ 2 & 0 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = 0$ $\quad 2\alpha = 0 \cdot \beta$, $\alpha = 0$, $\forall \beta$. $\Rightarrow$ пряма $x = 0$.



Ізокліна:

$$tg\alpha = \frac{dy}{dx} = 0$$

$$\frac{2x+y}{3x} = 0$$

$$y = -2x$$

$\lambda_1 = -1$, $\lambda_2 = -3$ — дійсні, різні, <0 і одного знаку, тобто вигляд точки — стійкий вузол.

Заміна: $\theta = -t$.

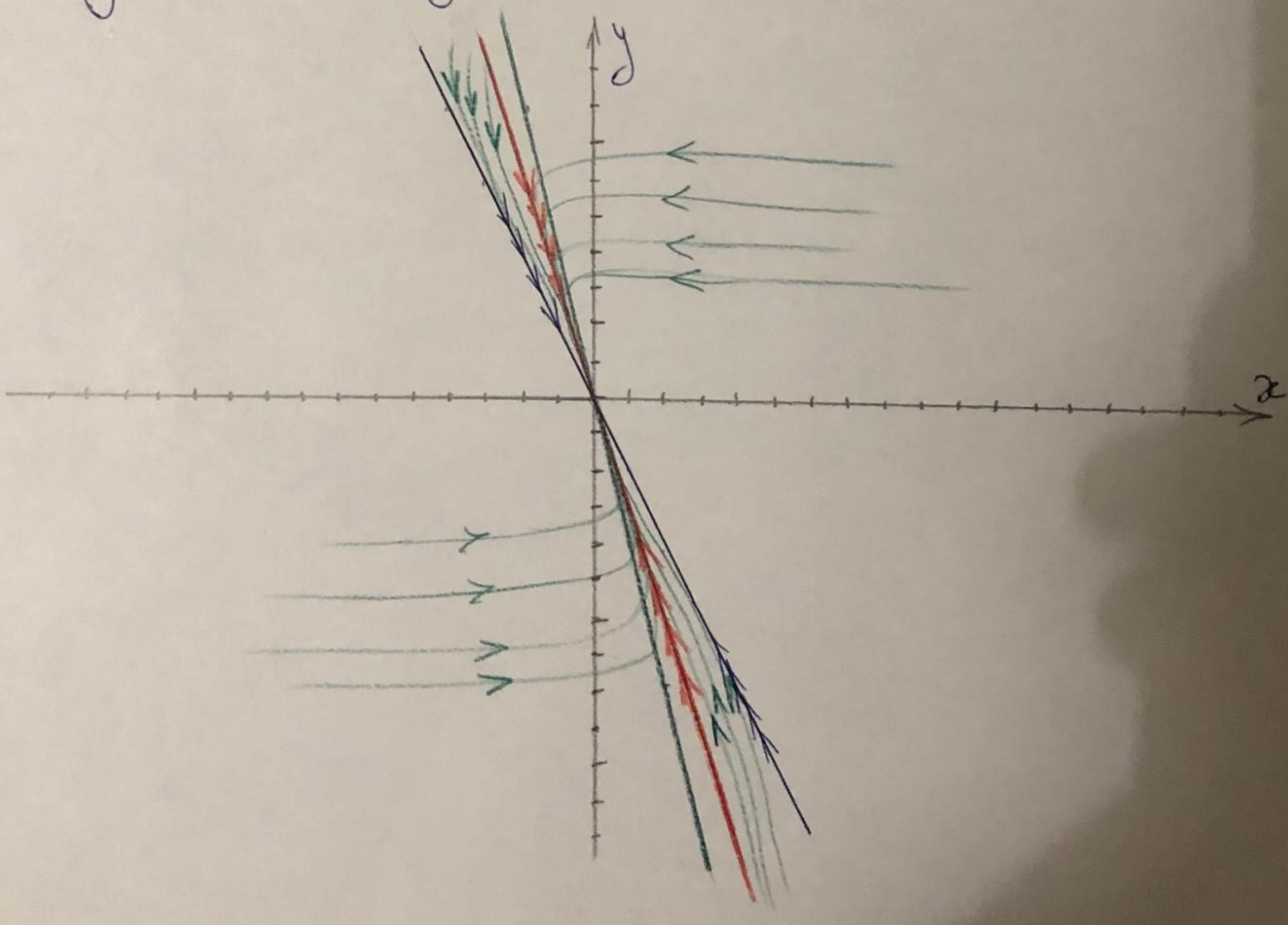
$$\begin{cases} \dot{x}(\theta) = 7x(\theta) + 2y(\theta) \\ \dot{y}(\theta) = -12x(\theta) - 3y(\theta) \end{cases} \qquad \lambda_1' = 3, \ \lambda_2' = 1.$$

$\lambda_1' = 3$: $\begin{pmatrix} 4 & 2 \\ -12 & -6 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \Rightarrow 2\alpha = -\beta$, $y = -2x$

$\lambda_2' = 1$ $\begin{pmatrix} 6 & 2 \\ -12 & -4 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \Rightarrow 3\alpha = -\beta$, $y = -3x$.

Ізокліна: $\operatorname{tg}\alpha = \frac{dy}{dx} = 0$. $\frac{12x + 3y}{-7x - 2y} = 0$.

$12x + 3y = 0 \Rightarrow y = -4x$.



$Ab = \begin{pmatrix} 3 & 0 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$. Відповідно, $S = (b, Ab) = \begin{pmatrix} 1 & 3 \\ -1 & 1 \end{pmatrix}$

$\det S = 1 - (-3) = 4 \neq 0 \Rightarrow$ система цілком керована

$$S^{-1} = \left(\begin{matrix} 1 & 3 \\ -1 & 1 \end{matrix}\middle|\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}\right) \sim \left(\begin{matrix} 1 & 3 \\ 0 & 4 \end{matrix}\middle|\begin{matrix} 1 & 0 \\ 1 & 1 \end{matrix}\right) \sim \left(\begin{matrix} 1 & 3 \\ 0 & 1 \end{matrix}\middle|\begin{matrix} 1 & 0 \\ 1/4 & 1/4 \end{matrix}\right) \sim \left(\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}\middle|\begin{matrix} \frac{1}{4} & -\frac{3}{4} \\ \frac{1}{4} & \frac{1}{4} \end{matrix}\right)$$

$$S^{-1} = \frac{1}{4} \begin{pmatrix} 1 & -3 \\ 1 & 1 \end{pmatrix}, \quad A^2 b = \begin{pmatrix} 3 & 0 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 9 \\ 7 \end{pmatrix}$$

Тоді $\begin{pmatrix} p_2 \\ p_1 \end{pmatrix} = -S^{-1} A^2 b = -\begin{pmatrix} \frac{1}{4} & -\frac{3}{4} \\ \frac{1}{4} & \frac{1}{4} \end{pmatrix} \begin{pmatrix} 9 \\ 7 \end{pmatrix} = \begin{pmatrix} 3 \\ -4 \end{pmatrix}$

За відомими значеннями власних чисел запишемо характеристичне рівняння:

$$(\lambda+1)(\lambda+3) = \lambda^2 + 4\lambda + 3$$

Знайдемо вектор стовпчик $(p-a)$:

$$(p-a) = \begin{pmatrix} -4 \\ 3 \end{pmatrix} - \begin{pmatrix} 4 \\ 3 \end{pmatrix} = \begin{pmatrix} -8 \\ 0 \end{pmatrix}$$

Остаточно обчислимо значення коефіцієнтів керування:

$$c = (S^{-1})^T \begin{pmatrix} 1 & 0 \\ p_1 & 1 \end{pmatrix}^{-1} (p-a) = \begin{pmatrix} 1/4 & 1/4 \\ -3/4 & 1/4 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -4 & 1 \end{pmatrix}^{-1} \begin{pmatrix} -8 \\ 0 \end{pmatrix} =$$

$$= \begin{pmatrix} 1/4 & 1/4 \\ -3/4 & 1/4 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix} \begin{pmatrix} -8 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/4 & 1/4 \\ -3/4 & 1/4 \end{pmatrix} \begin{pmatrix} -8 \\ -32 \end{pmatrix} = \begin{pmatrix} -10 \\ -2 \end{pmatrix}.$$

Отже, шукане модальне керування:

$u(t) = -10x(t) - 2y(t)$.

$$\dot{x} = 3x - 10x - 2y = -7x - 2y \qquad \tilde{A} = \begin{pmatrix} -7 & -2 \\ 12 & 3 \end{pmatrix}.$$
$$\dot{y} = 2x + y + 10x + 2y = 12x + 3y$$

$\det(\tilde{A} - \lambda E) = \lambda^2 + 4\lambda + 3 = 0 \quad \lambda_1 = -1, \ \lambda_2 = -3.$

Код програми для розімкненої системи

```
# Розімкнена система

# Коефіцієнти системи
# x' = m11 * x + m12 * y
# y' = m21 * x + m22 * y
m11 = 3; m12 = 0;
m21 = 2; m22 = 1;

# Матриця з коефіцієнтів
M = matrix([[m11, m12], [m21, m22]]);

# Власні вектори
eigenVectors = M.eigenvectors_right();
v1 = eigenVectors[0][1][0];
v2 = eigenVectors[1][1][0];

x, y, t = var('x y t');

# x1 і y1 для вирішення системи диф-рівнянь
x1 = function('x1')(t);
y1 = function('y1')(t);

# Сепаратриси
if (v1[0] != 0):
    separ1 = solve([v1[0] * y == v1[1] * x], y)[0].right();
else:
    separ1 = (x == 0);

if (v2[0] != 0):
    separ2 = solve([v2[0] * y == v2[1] * x], y)[0].right();
else:
    separ2 = (x == 0);
```

```
# Ізокліна
isocline = solve([(m21 * x + m22 * y)/(m11 * x + m12 * y)], y)[0].right();

# Сама система диф-рівнянь
dx = diff(x1, t) == m11 * x1 + m12 * y1;
dy = diff(y1, t) == m21 * x1 + m22 * y1;

# Малюємо графік розміром (plotRange; plotRange)
plotRange = 25;

# Малюємо поле напрямків
plt = plot_vector_field([m11 * x + m12 * y, m21 * x + m22 * y],
                        [x, -plotRange, plotRange],
                        [y, -plotRange, plotRange],
                        axes_labels=['$x$','$y(x)$']);

# Малюємо сепаратриси
if (v1[0] != 0):
    plt += plot(separ1, (-plotRange, plotRange), color = "blue", thickness = 2);
else:
    plt += implicit_plot(separ1, (x, -plotRange, plotRange), (y, -plotRange, plotRange),
color = "blue", axes = true, linewidth = 3);

if (v2[0] != 0):
    plt += plot(separ2, (-plotRange, plotRange), color = "red", thickness = 2);
else:
    plt += implicit_plot(separ2, (x, -plotRange, plotRange), (y, -plotRange, plotRange),
color = "red", axes = true, linewidth = 3);

# Малюємо ізокліну
plt += plot(isocline, (-plotRange/2, plotRange/2), color = "black", thickness = 3);

for i in range (-10, 10):
```

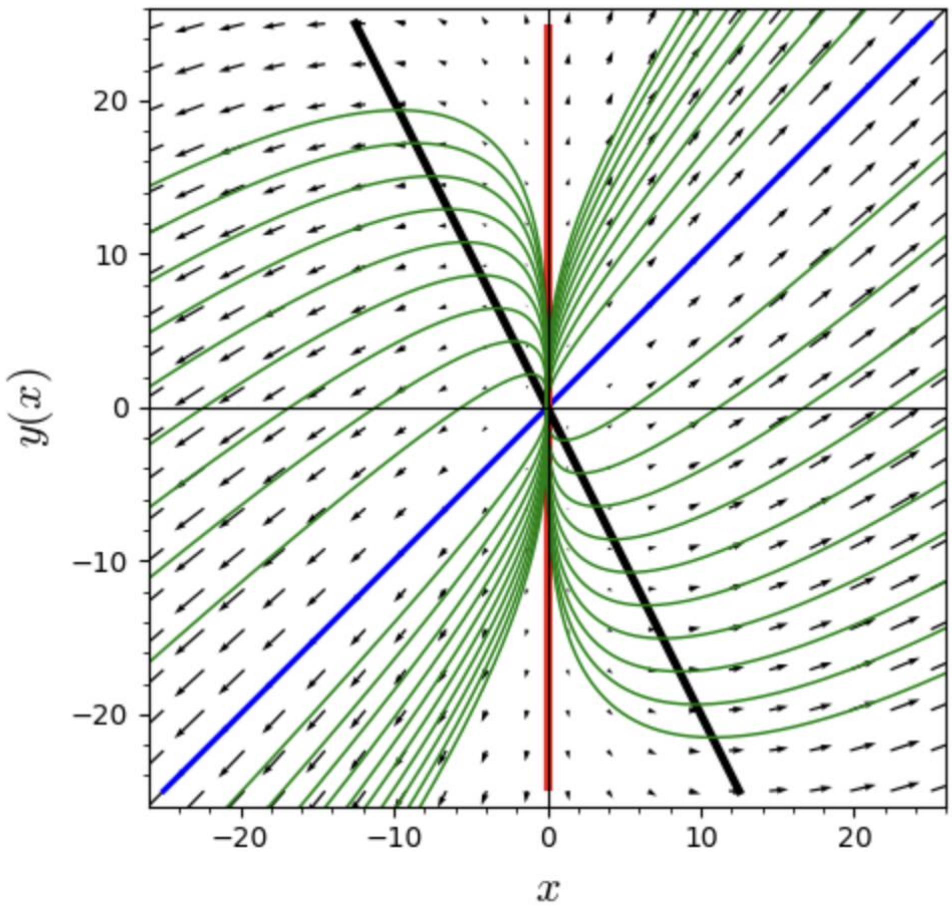
```
    # Знаходимо черговий розв'язок задачі Коші
    x_t, y_t = desolve_system([dx, dy], [x1, y1], ics = [0, -i*4, i]);
    # Малюємо роз'язок задачі Коші
    plt += parametric_plot((x_t.rhs(), y_t.rhs()), (t, -plotRange, plotRange), color =
"green");

for i in range (-10, 10):
    # Знаходимо черговий розв'язок задачі Коші
    x_t, y_t = desolve_system([dx, dy], [x1, y1], ics = [0, i/2, i*2]);
    # Малюємо роз'язок задачі Коші
    plt += parametric_plot((x_t.rhs(), y_t.rhs()), (t, -plotRange, plotRange), color =
"green");

# Показати графік
plt.show(xmin = -plotRange, xmax = plotRange, ymin =-plotRange, ymax =
plotRange);
```

Код програми для розімкненої системи

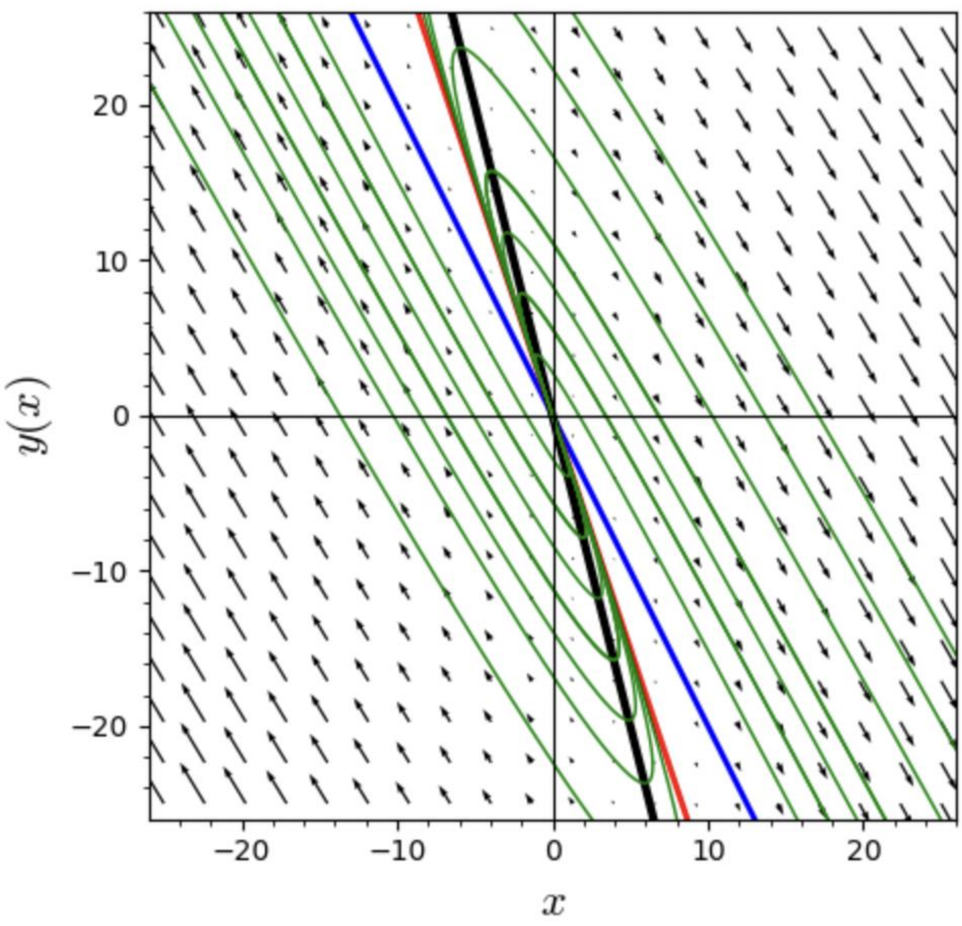
```
# Замкнена система

# Коефіцієнти системи
# x' = m11 * x + m12 * y
# y' = m21 * x + m22 * y
m11 = 7; m12 = 2;
m21 = -12; m22 = -3;

# Матриця з коефіцієнтів
M = matrix([[m11, m12], [m21, m22]]);

# Власні вектори
eigenVectors = M.eigenvectors_right();
v1 = eigenVectors[0][1][0];
v2 = eigenVectors[1][1][0];

x, y, t = var('x y t');

# x1 і y1 для вирішення системи диф-рівнянь
x1 = function('x1')(t);
```

```
y1 = function('y1')(t);

# Сепаратриси
if (v1[0] != 0):
    separ1 = solve([v1[0] * y == v1[1] * x], y)[0].right();
else:
    separ1 = (x == 0);

if (v2[0] != 0):
    separ2 = solve([v2[0] * y == v2[1] * x], y)[0].right();
else:
    separ2 = (x == 0);

# Ізокліна
isocline = solve([(m21 * x + m22 * y)/(m11 * x + m12 * y)], y)[0].right();

# Сама система диф-рівнянь
dx = diff(x1, t) == m11 * x1 + m12 * y1;
dy = diff(y1, t) == m21 * x1 + m22 * y1;

# Малюємо графік розміром (plotRange; plotRange)
plotRange = 25;



# Малюємо поле напрямків
plt = plot_vector_field([m11 * x + m12 * y, m21 * x + m22 * y],
                        [x, -plotRange, plotRange],
                        [y, -plotRange, plotRange],
                        axes_labels=['$x$','$y(x)$']);

# Малюємо сепаратриси
if (v1[0] != 0):
    plt += plot(separ1, (-plotRange, plotRange), color = "blue", thickness = 2);
else:
```

```
    plt += implicit_plot(separ1, (x, -plotRange, plotRange), (y, -plotRange, plotRange),
color = "blue", axes = true, linewidth = 3);

if (v2[0] != 0):
    plt += plot(separ2, (-plotRange, plotRange), color = "red", thickness = 2);
else:
    plt += implicit_plot(separ2, (x, -plotRange, plotRange), (y, -plotRange, plotRange),
color = "red", axes = true, linewidth = 3);

# Малюємо ізокліну
plt += plot(isocline, (-plotRange/2, plotRange/2), color = "black", thickness = 3);

for i in range (-5, 5):
    # Знаходимо черговий розв'язок задачі Коші
    x_t, y_t = desolve_system([dx, dy], [x1, y1], ics = [0, -i*4, i]);
    # Малюємо роз'язок задачі Коші
    plt += parametric_plot((x_t.rhs(), y_t.rhs()), (t, -plotRange, plotRange), color =
"green");

for i in range (-5, 5):
    # Знаходимо черговий розв'язок задачі Коші
    x_t, y_t = desolve_system([dx, dy], [x1, y1], ics = [0, i/2, i*2]);
    # Малюємо роз'язок задачі Коші
    plt += parametric_plot((x_t.rhs(), y_t.rhs()), (t, -plotRange, plotRange), color =
"green");

# Показати графік
plt.show(xmin = -plotRange, xmax = plotRange, ymin =-plotRange, ymax =
plotRange);
```



20
10
0
−10
−20
$y(x)$
−20
−10
0
10
20
x



20
10
0
−10
−20
y(x)
−20
−10
0
10
20
x