МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Національний технічний університет
«Харківський політехнічний інститут»
Кафедра КМАД

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №2

Виконав студент:

Омельніцький Андрій Миколайович
Група: КН-120А

Перевірила:

Ольга Василівна Костюк

Харків, 2023 г.

Зміст

Глава 1

Мета роботи

Побудова математичної моделі Вольтерри, дослідження моделі із використанням комп'ютерного моделювання.

Порядок виконання:

1. Знайти стаціонарні точки системи диференціальних рівнянь для заданих значень параметрів.
2. Розв'язати систему, використовуючи чисельні методи розв'язання диференціальних рівнянь та отримати залежності кількості жертв та хижаків від часу для заданих параметрів моделі.
3. Вивести графіки розв'язків у часі та у фазовому просторі для початкових значень x0, y0.
4. Виконати моделювання й оцінити поведінку системи за різних значень параметрів, а також за різних початкових умов. При цьому необхідно врахувати певні обмеження для можливих значень параметрів, спираючись на змістовне наповнення моделі. Надати рекомендації щодо співвідношень параметрів моделі.
5. Порівняти результати моделювання з результатами, отриманими в попередній лабораторній роботі.
6. Усі результати, отримані в ході виконання роботи, занести до звіту. Зробити висновки.

Глава 2

Аналіз моделі

Наведемо модель та її параметри.

$$\begin{cases} \dfrac{\partial x}{\partial t} = \varepsilon_1 x - \gamma_{12} xy - \gamma_{11} x^2 \\ \dfrac{\partial y}{\partial t} = \gamma_{21} xy - \varepsilon_2 y - \gamma_{22} y^2 \end{cases}$$

Значення параметрів моделі: $\varepsilon_1 = 2.1; \gamma_{12} = 0.51; \gamma_{11} = 0.1; \gamma_{21} = 0.67; \varepsilon_2 = 3; \gamma_{22} = 0.13; x_0 = 1.9; y_0 = 1.6;$

2.1. Пункт 1

Визначемо стаціонарні точки. Для цього треба розвязати систему при $\frac{\partial x}{\partial t} = 0, \frac{\partial y}{\partial t} = 0.$

$$\begin{cases} \varepsilon_1 x - \gamma_{12} xy - \gamma_{11} x^2 = 0 \\ \gamma_{21} xy - \varepsilon_2 y - \gamma_{22} y^2 = 0 \end{cases}$$

Рішенням цієї системи будуть такі стаціонарні точки:

а)

$$x_0 = 0, y_0 = 0;$$

б)

$$x_0 = 0, y_0 = -\frac{\varepsilon_2}{\gamma_{22}};$$

в)

$$x_0 = \frac{\varepsilon_1}{\gamma_{11}}, y_0 = 0;$$

г)

$$x_0 = \frac{\varepsilon_1 \gamma_{22} + \varepsilon_2 \gamma_{12}}{\gamma_{11}\gamma_{22} + \gamma_{12}\gamma_{21}}, y_0 = \frac{\varepsilon_1 \gamma_{21} - \varepsilon_2 \gamma_{11}}{\gamma_{11}\gamma_{22} + \gamma_{12}\gamma_{21}};$$

Проаналізуємо результати:
Рішення а) відповідає відсутності жертв і хижаків, і тому нецікаве.
Рішення б) фізично неможливе, оскільки число хижаків може бути від'ємним.
Рішення в) відповідає динамічній рівновазі чисельності жертв без хижаків.
Рішення г) найцікавіше для подальшого аналізу.

Тому порахуемо точки в) та г). Точку в) позначемо як $P_1(21, 0)$. Точку г) позначемо як $P_2(5.08317, 3.12095)$.

2.2. Пункт 2

1.9 y0 = 1.6 Розв'яжемо систему та зобразимо залежність кількості жертв та хижаків з часом для $x_0 = 1.9, y_0 = 1.6$.

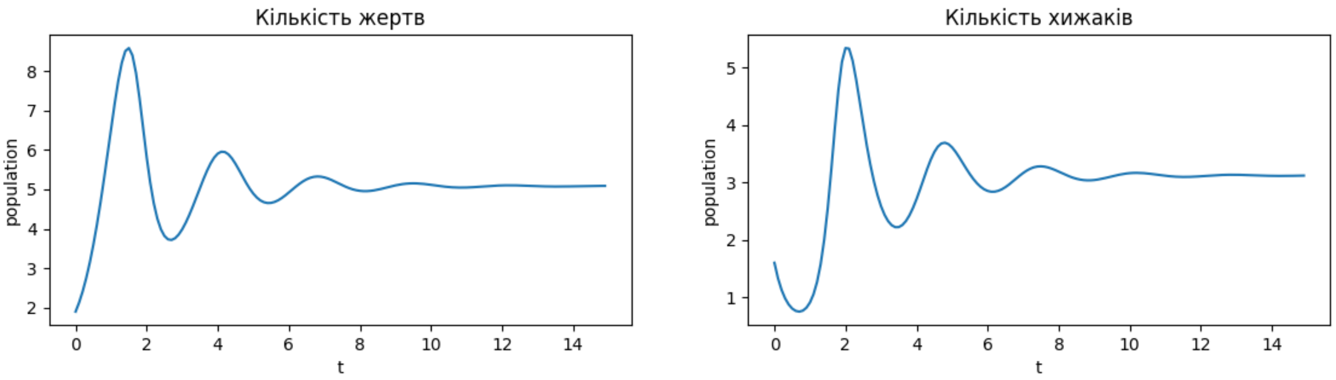

Рис. 2.1. Залежність кількості хижаків та жертв від часу

2.3. Пункт 3

Зобразимо графіки розв'язків у фазовому просторі для початкових значень $x_0 = 1.9, y_0 = 1.6$.

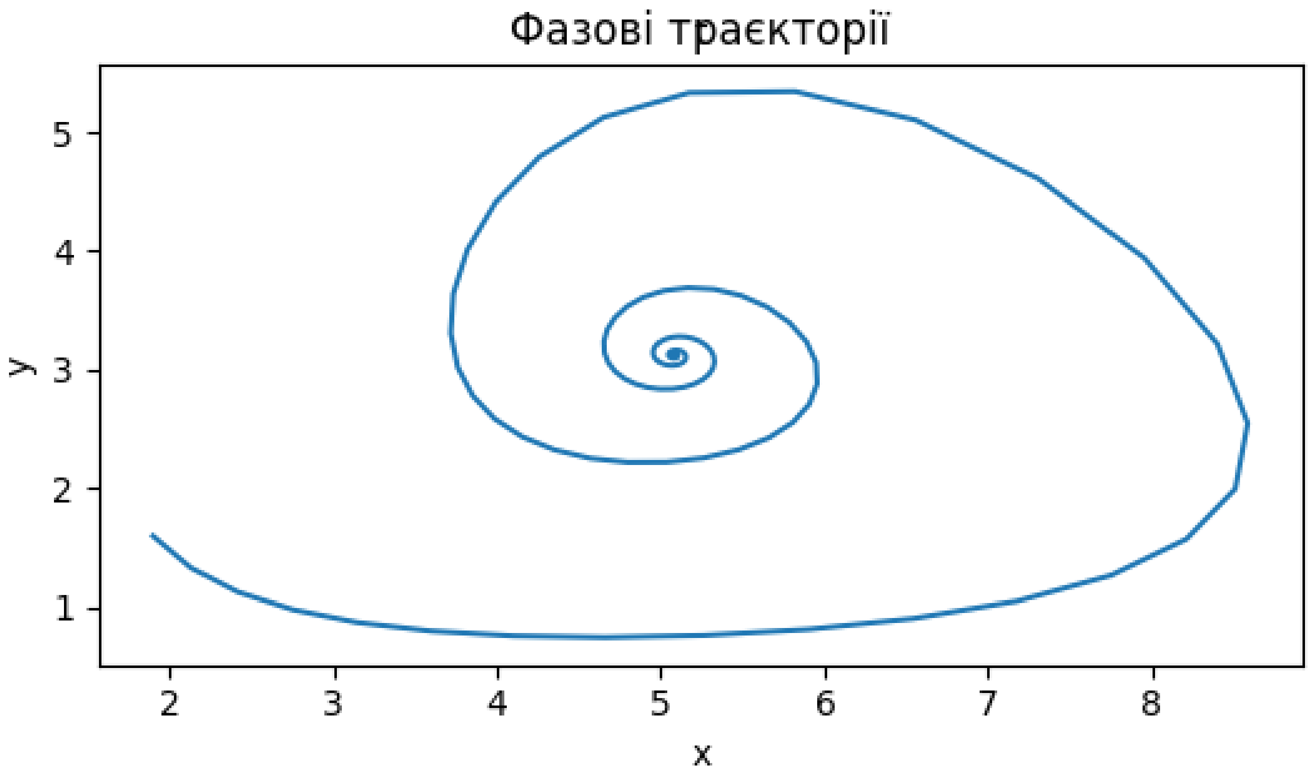

Рис. 2.2. Фазові траєкторії

2.4. Пункт 4

2.4.1. Основна частина

Розглянемо поведінку моделі за різніх початковіх значень.

2.4.2. Приклад 1

Розглянемо критичку точку P1 з П.1. Тоді $x_0 = 21; y_0 = 0;$

Як можно побачити чисельність популяції жертви не змінюється. А популяція хижаків не змінюється, бо з самого початку не було хижаків.

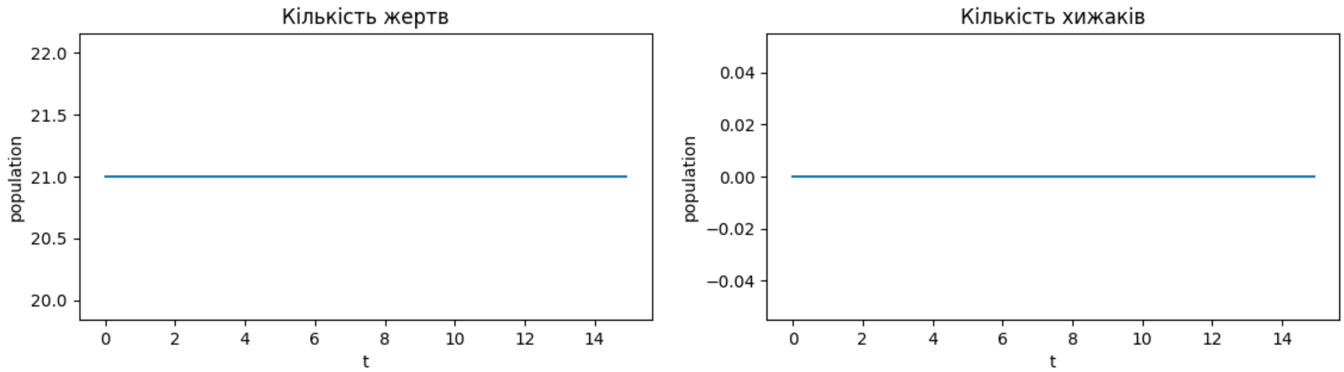

Рис. 2.3. $x_0 = 21; y_0 = 0;$

2.4.3. Приклад 2

Розглянемо іньшу критичку точку P2 з П.1. Тоді $x_0 = 5.08317; y_0 = 3.12095;$

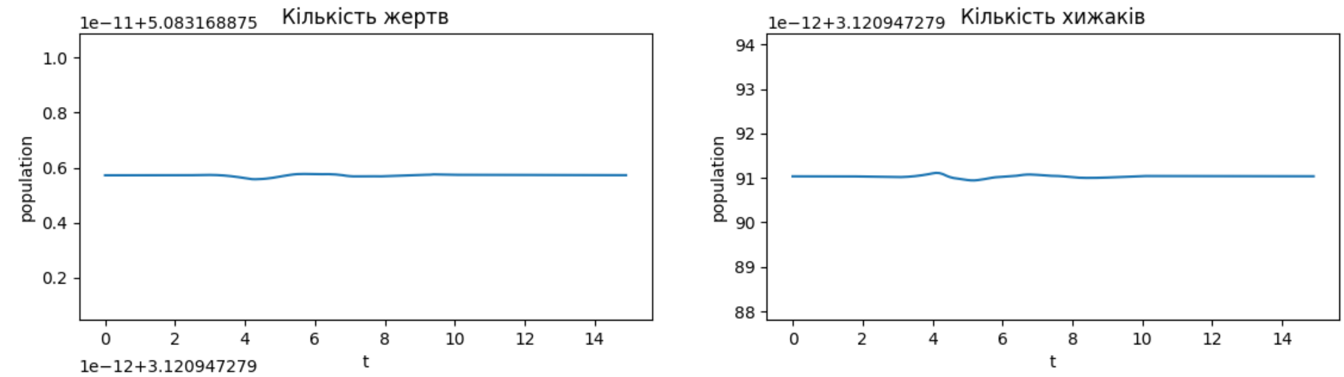

Рис. 2.4. $x_0 = 5.08317; y_0 = 3.12095;$

Як можно побачити чисельність популяцій жертви та хижаків залишається приблизно на рівні початковіх значень.

2.4.4. Приклад 3

Розглянемо систему за таких параметрів $(x_0 = 100; y_0 = 0;)$, $(x_0 = 2; y_0 = 0;)$, $(x_0 = 0; y_0 = 0;)$

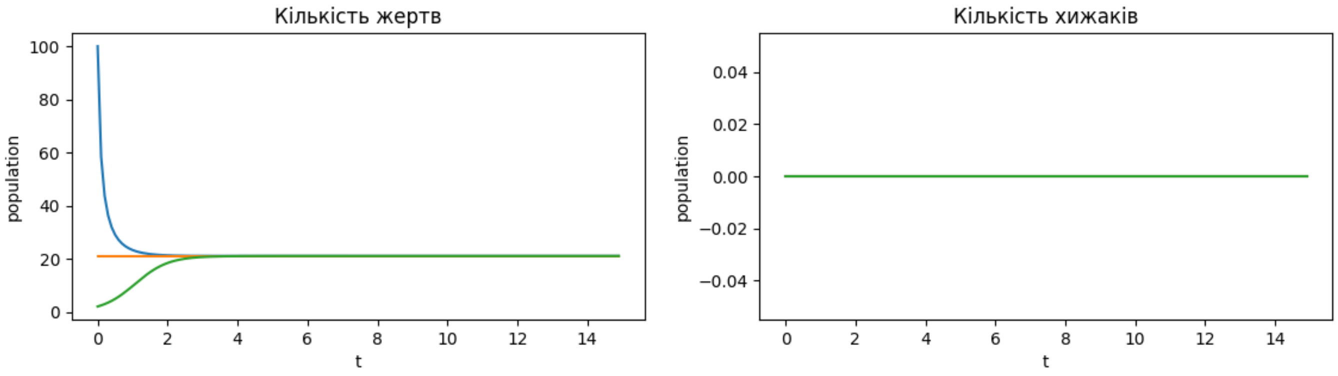

Рис. 2.5. Приклад 3

Можна побачити, що популяція жертв справді перебувая в рівновазі на значенні $x = \frac{\varepsilon_1}{\gamma_{11}}$, якщо хижаків нема.

2.4.5. Приклад 4

Розглянемо систему за таких параметрів $x_0 = 1.9; y_0 = 1.6; \varepsilon_1 = 10$

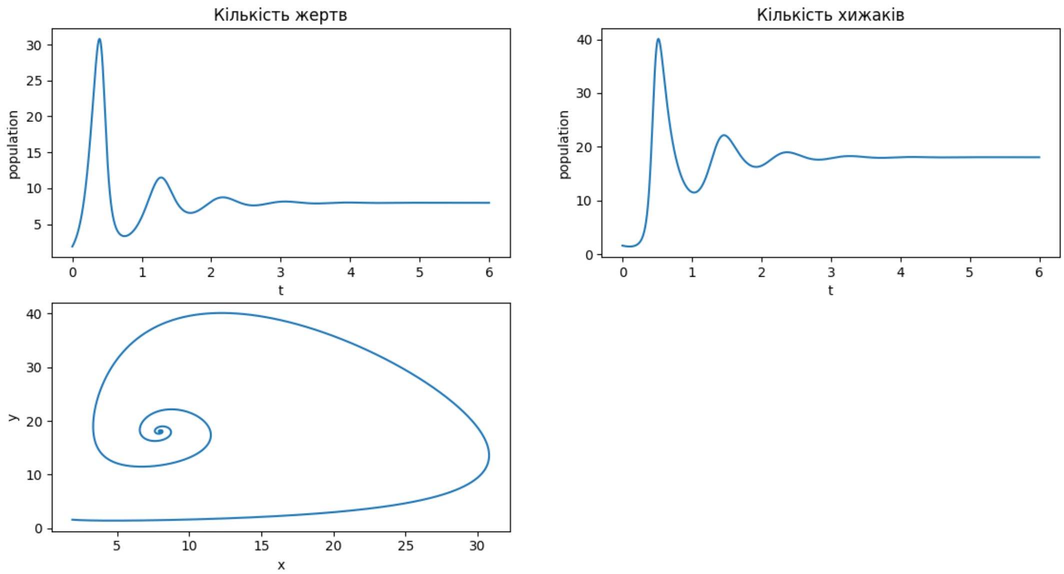

Рис. 2.6. Приклад 4

2.4.6. Приклад 5

Розглянемо систему за таких параметрів $x_0 = 1.9; y_0 = 1.6; \varepsilon_2 = 10$

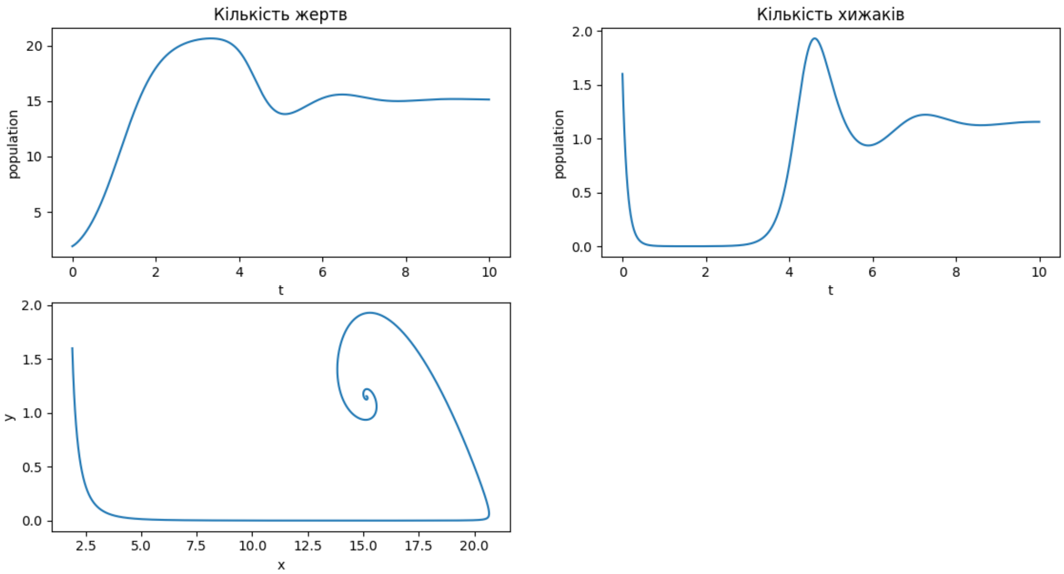

Рис. 2.7. Приклад 5

2.5. Пункт 5

Данна модель на відміну від попередньої за допомогую додаткових параметрів дозволяє обмежити зростання популяцій. Тобто ми можемо під час моделювання процеса явно задати обмеження популяцій. Як приклад можно навести результати які були продемонстровані у прикладі 3 П.4. Де популяція жертв за вісдсутністю хижаків з часом прагне до $x = \frac{\varepsilon_1}{\gamma_{11}}$.

Глава 3

Висновок

У ході лабораторної роботи була побудована і досліджена математична модель відносин Вольтерри.

Глава 4

Код програми

4.1. main.py

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint, solve_ivp


e1 = 2.1
y12 = 0.51
y11 = 0.1
y21 = 0.67
e2 = 3
y22 = 0.13

# e1 = 2.1; y12 = 0.51; y11 = 0.1; y21 = 0.67; e2 = 3; y22 = 0.13

# x0 = 1.9
# y0 = 1.6

x0 = 1.9
y0 = 1.6

# e2 = 10

t = np.arange(0, 10, 0.001)

def f(t, y):
    result = [
        e1*y[0] - y12*y[0]*y[1] - y11*(y[0]**2),
        y21*y[0]*y[1] - e2*y[1] - y22*(y[1]**2),
    ]
    return np.array(result)


```

```
def main():
    p1 = (e1 / y11, 0)
    print(f'P1 ({e1 / y11:.5f}, 0)')

    t_x = (e1 * y22 + e2 * y12) / (y11 * y22 + y12 * y21)
    t_y = (e1 * y21 - e2 * y11) / (y11 * y22 + y12 * y21)
    # x0, y0 = t_x, t_y
    p2 = (t_x, t_y)
    print(f'P2 ({t_x:.5f}, {t_y:.5f})')

    y1 = odeint(f, [x0, y0], t, tfirst=True)


    y2 = odeint(f, [21, 0], t, tfirst=True)
    y3 = odeint(f, [2, 0], t, tfirst=True)

    # print(f'min={np.min(y1[:, 0]):.5f} max={np.max(y1[:, 0]):.5f}')

    ax1 = plt.subplot(2, 2, 1)
    ax2 = plt.subplot(2, 2, 2)
    ax3 = plt.subplot(2, 2, 3)

    ax1.set_title('                 ')
    ax1.set_xlabel('t')
    ax1.set_ylabel('population')
    ax2.set_title('                      ')
    ax2.set_xlabel('t')
    ax2.set_ylabel('population')
    # ax3.set_title('                         ')
    ax3.set_xlabel('x')
    ax3.set_ylabel('y')

    ax1.plot(t, y1[:, 0], label='1')
    ax2.plot(t, y1[:, 1], label='2')

    # ax1.plot(t, y2[:, 0], label='1')
    # ax2.plot(t, y2[:, 1], label='2')
    # ax1.plot(t, y3[:, 0], label='1')
    # ax2.plot(t, y3[:, 1], label='2')
    ax3.plot(y1[:, 0], y1[:, 1], label='3')
```

```
    plt.show()

if __name__ == '__main__':
    main()
```