МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Національний технічний університет
«Харківський політехнічний інститут»
Кафедра КМАД

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №1

Виконав студент:

Омельніцький Андрій Миколайович
Група: КН-120А

Перевірила:

Ольга Василівна Костюк

Харків, 2023 г.

Зміст

Глава 1

Мета роботи

Побудова математичної моделі відносин “хижак–жертва”, дослідження моделі із використанням комп’ютерного моделювання.

Порядок виконання:

1. Знайти стаціонарні точки значення чисельності жертв та хижаків для системи рівнянь моделі Вольтерра-Лотки.
2. Розв‘язати систему, використовуючи чисельні методи розв’язання диференціальних рівнянь.
3. Вивести графіки розв‘язків у часі та у фазовому просторі.
4. Знайти максимальну та мінімальну кількості хижаків та жертв.
5. Знайти періоди коливань чисельності хижаків та жертв.
6. Виконати моделювання й оцінити поведінку системи за різних значень параметрів α, β, ϵ, δ, а також за різних початкових умов x_0 та y_0.
7. Дослідити, при яких відхиленнях від стаціонарних значень чисельності гармонічні коливання змінюються складними коливаннями, а форма фазової траєкторії перестає бути еліпсом.
8. Знайти співвідношення параметрів, у яких відбувається природне зростання популяцій.
9. Отримані результати занести до звіту, зробити висновки.

Глава 2

Аналіз моделі

Наведемо модель та її параметри.

$$\begin{cases} \dfrac{dx}{dt} = \epsilon x - \alpha xy \\ \dfrac{dx}{dt} = \delta xy - \beta y \end{cases}$$

Значення параметрів моделі: $\epsilon = 1.2; \beta = 0.35; \alpha = 0.02; \delta = 0.0001$

2.1. Пункт 1

Визначемо стаціонарні точки. Для цього треба розвязати систему при $\frac{dx}{dt} = 0, \frac{dy}{dt} = 0$.

$$\begin{cases} \epsilon x - \alpha xy = 0 \\ \delta xy - \beta y = 0 \end{cases}$$

Вирішимо за домомогою WolframAlpha. Отримаємо значення стаціонарних точок системи: $P1(x = 0, y = 0); P2(x = 3500, y = 60)$

2.2. Пункт 2

Розв'яжемо систему та зобразимо залежність кількості жертв та хижаків з часом для $x_0 = 500, y_0 = 100$.

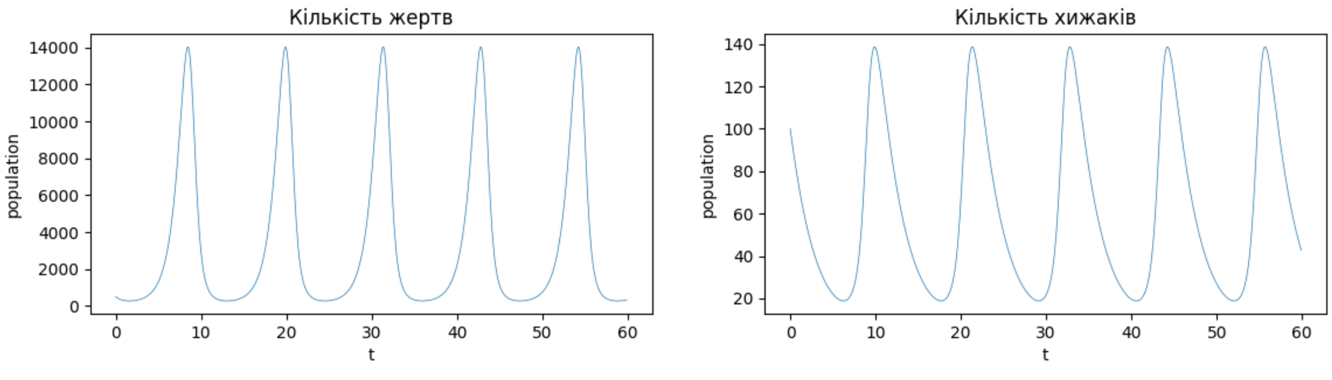

Рис. 2.1. Залежність кількості хижаків та жертв від часу

2.3. Пункт 3

Зобразимо графіки фазових кривих для вихідних кількостей жертв та хижаків (500, 100), (100, 100), (100, 200).

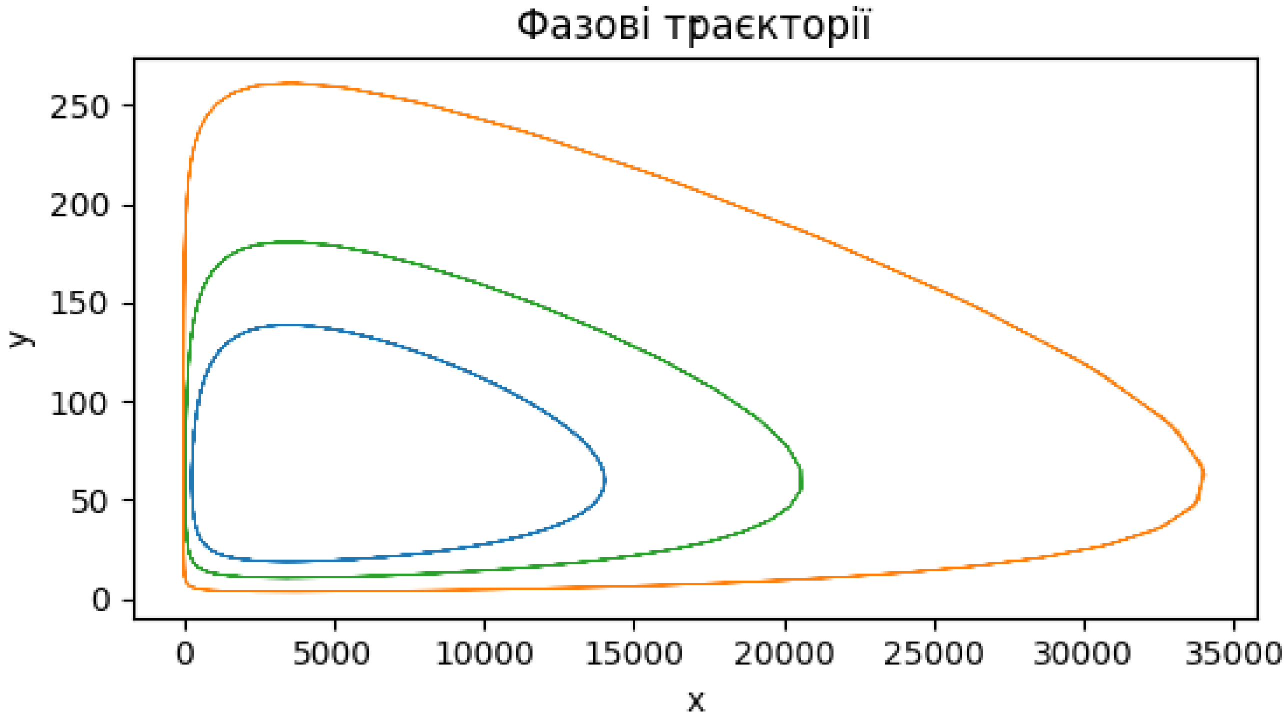

Рис. 2.2. Фазові траєкторії

2.4. Пункт 4

Знайдемо максимальну та мінімальну кількість жертв та хижаків для початкового чтану (500, 100).
Кількість жертв: $min = 274.77421; max = 14043.55672$
Кількість хижаків: $min = 18.81215; max = 138.66362$

2.5. Пункт 5

Знайдемо періоди коливань чисельності хижаків та жертв для початкового чтану (500, 100).
Знайдемо період для жертв. Для цього порахуємо у який час кількість жертв є максимальною.
Результат обчислень:

i t val
1 8.5 14029.13259
2 19.9 14043.55672
3 31.3 14022.86070
4 42.8 14043.34374
5 54.2 14031.03618
6 65.7 14041.02555
7 77.1 14037.21231
8 88.6 14036.58130
9 100.0 14041.36077
10 111.5 14029.98858
11 122.9 14043.45647

Терпер можемо порахувати період $T_{prey} \approx 11.44$.

Знайдемо період для хижаків. Для цього порахуємо у який час кількість хижаків є максимальною.
Результат обчислень:

i t val
1 9.9 138.66362
2 21.3 138.62480
3 32.8 138.65011
4 44.2 138.64854
5 55.7 138.62792
6 67.1 138.66298
7 78.6 138.59722
8 90.0 138.66828
9 101.4 138.58739
10 112.9 138.66459
11 124.3 138.62165

Терпер можемо порахувати період $T_{predator} \approx 11.44$.
Можемо побачити, що періоди коливань кількості хижаків та жертв дорівнюють: $T \approx 11.44$.

2.6. Пункт 6

Приклад з параметрами $x_0 = 10000; y_0 = 100; \alpha = 0.1$

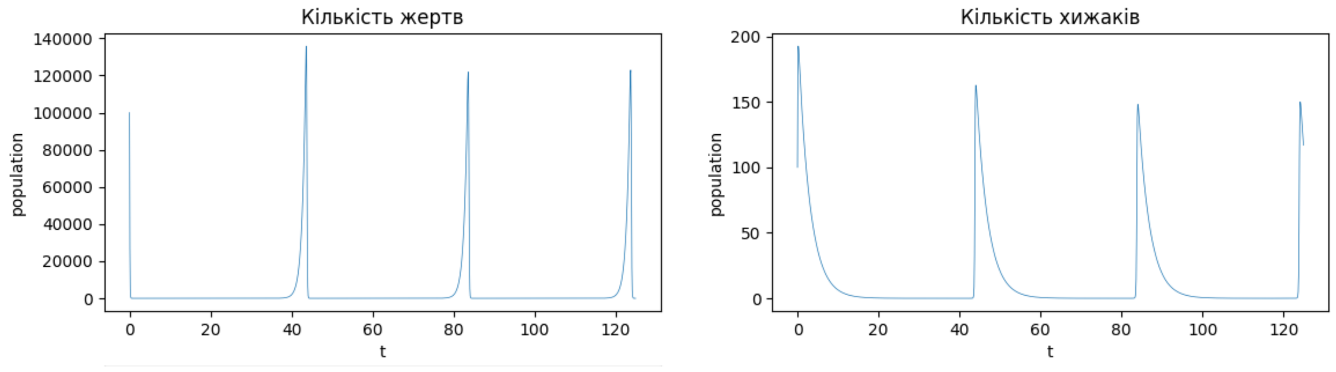

Рис. 2.3. Привклад 1

Приклад з параметрами $x_0 = 500; y_0 = 100; \epsilon = 0.001$

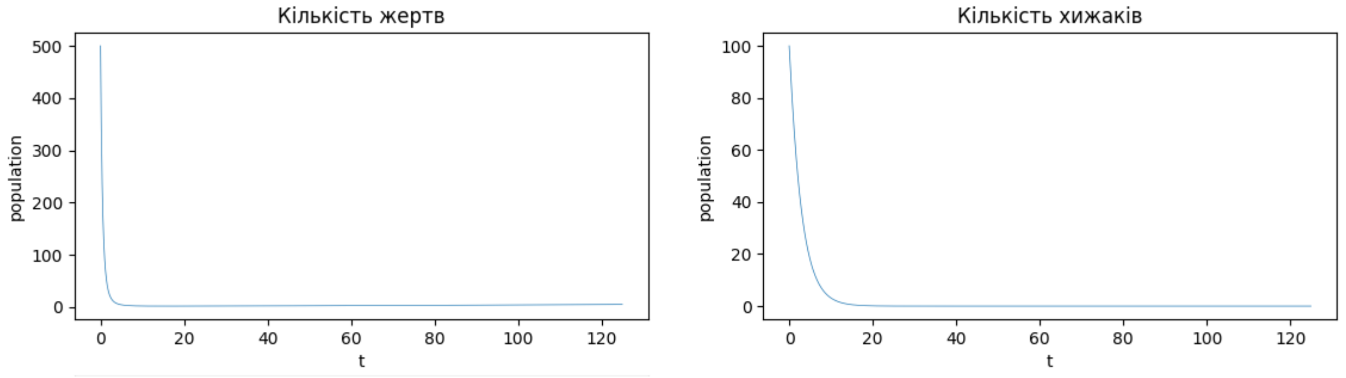

Рис. 2.4. Привклад 1

Приклад з параметрами $x_0 = 10000; y_0 = 100; \beta = 0.8$

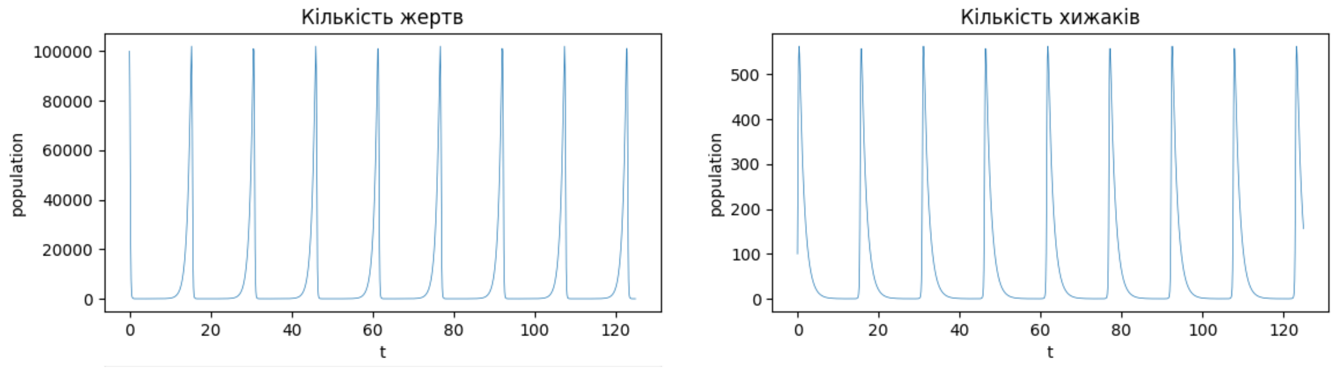

Рис. 2.5. Привклад 1

2.7. Пункт 7

Розглянемо стаціонарну точку P2(3500, 60) з П.1. Можемо побачити, що чисельність популяції є сталою.

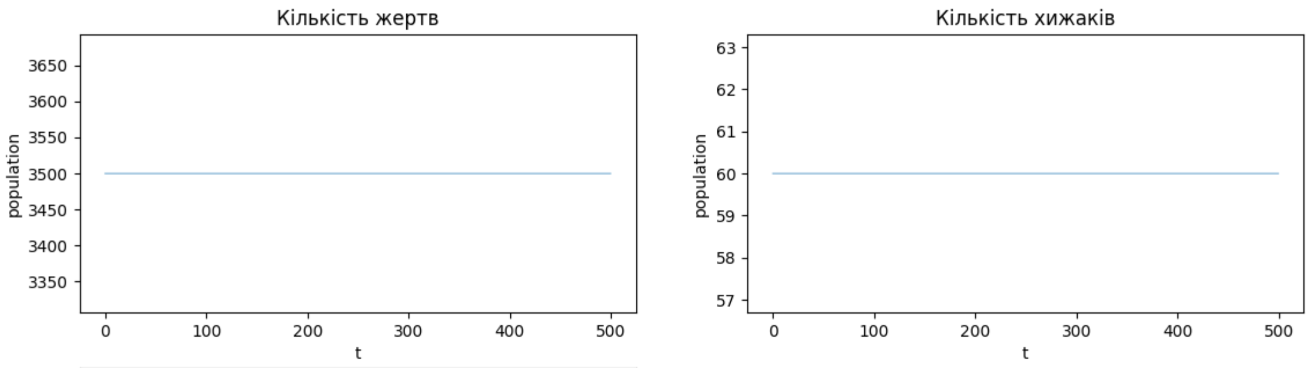

Рис. 2.6. $x_0 = 3500; y_0 = 60;$

При $x_0 \in [3000, 4000]$ та $y_0 \in [55, 65]$ фазова траєкторія має форму еліпсу. Як приклад розглянемо $x_0 = 3800; y_0 = 63;$.

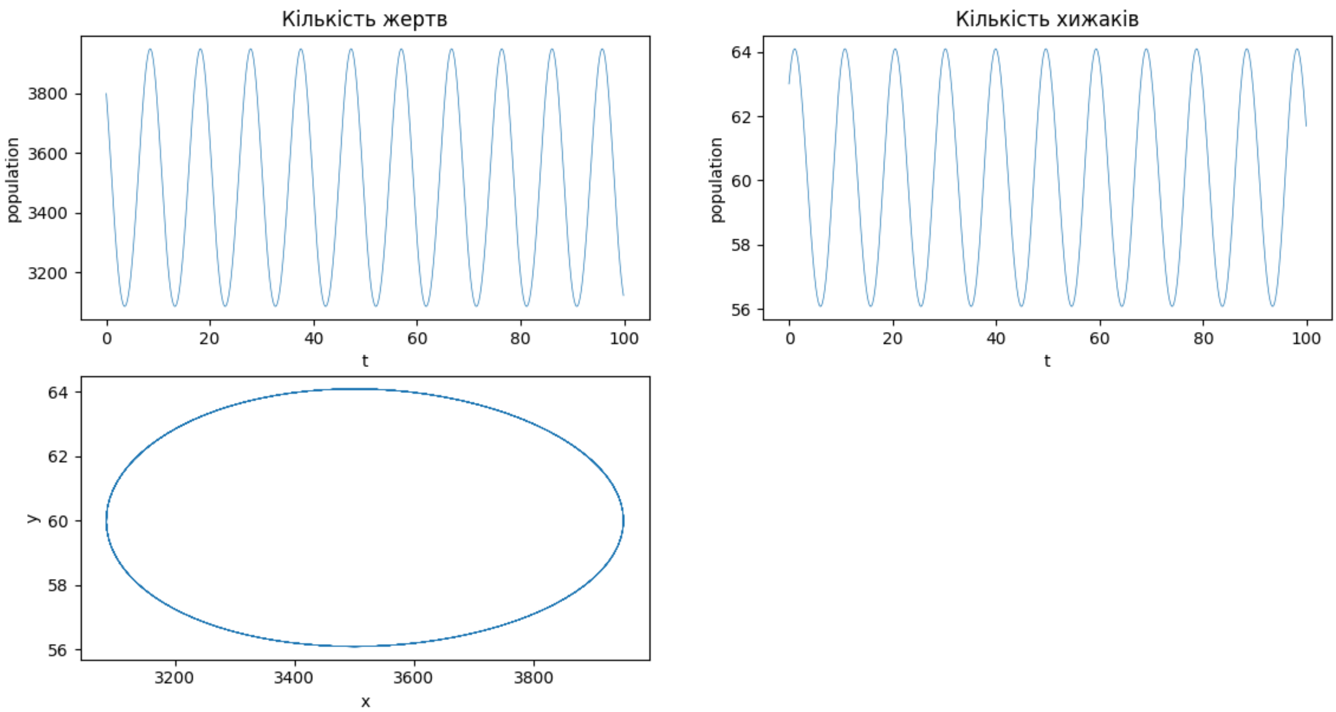

Рис. 2.7. $x_0 = 3800; y_0 = 63;$

Розглянемо стаціонарну точку P1(0, 0) з П.1. Можемо побачити, що якщо змінити тільки x_0, то будемо мати зростання популяції жертв див. рис. 2.8. А якщо змінити тільки y_0, то будемо мати вимирання популяції хижаків див. рис. 2.9.

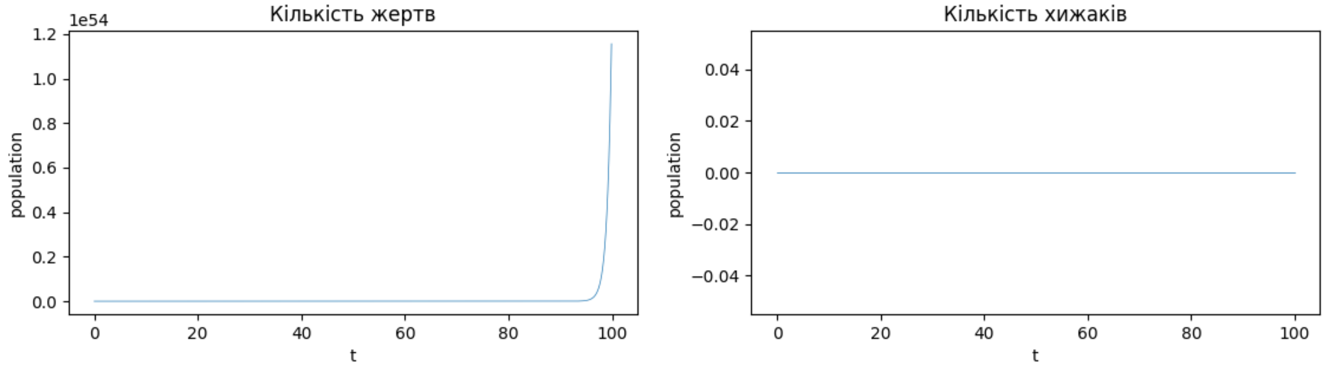

Рис. 2.8. $x_0 = 100; y_0 = 0;$

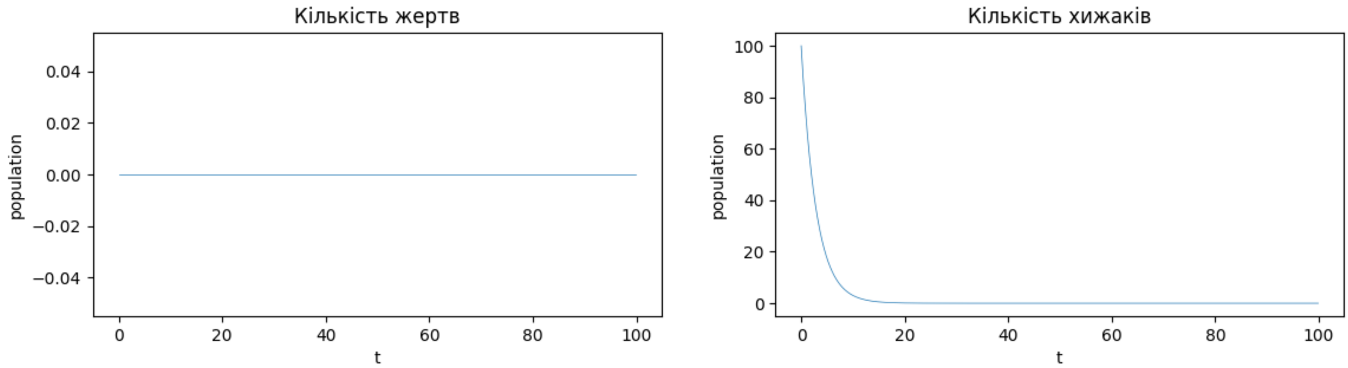

Рис. 2.9. $x_0 = 0; y_0 = 100;$

2.8. Пункт 8

Природне зростання популяції може бути тільки якщо хижаків немає. Як приклад розглянемо $x_0 = 300; y_0 = 0;$.

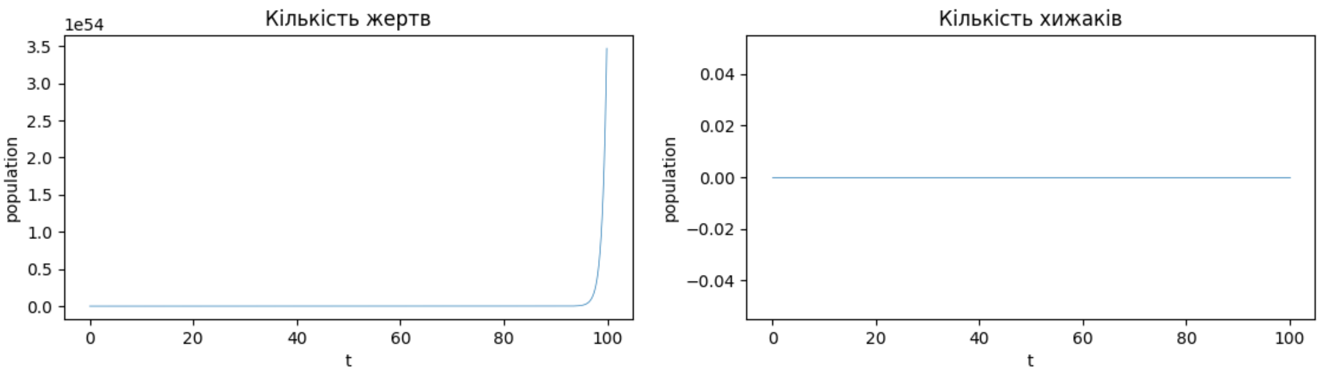

Рис. 2.10. $x_0 = 300; y_0 = 0;$

Глава 3

Висновок

У ході лабораторної роботи була побудована і досліджена математична модель відносин “хижак–жертва”.

Глава 4

Код програми

4.1. main.py

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

e = 1.2
b = 0.35
a = 0.02
d = 0.0001

# e = 0.01

# a = 0.1

# b = 0.8

# x0 = 0
# y0 = 0

x0 = 500
y0 = 100

# x0 = 3500
# y0 = 60

# x0 = 3000
# x0 = 4000

# y0 = 55
# y0 = 65


# x0 = 300
# y0 = 0
```

```


t_step = 0.1
t_start = 0
t_end = 100
t = np.arange(t_start, t_end, t_step)

def f(t, y):
    result = [
        y[0] * e - a * y[0] *y[1],
        d * y[1] * y[0] - y[1] * b,
    ]
    return np.array(result)


def main():
    y1 = odeint(f, [x0, y0], t, tfirst=True)
    # print(y1)

    y2 = odeint(f, [100, 200], t, tfirst=True)
    y3 = odeint(f, [100, 100], t, tfirst=True)

    print('  :')
    print(f'min={np.min(y1[:, 0]):.5f} max={np.max(y1[:, 0]):.5f}')
    print('  :')
    print(f'min={np.min(y1[:, 1]):.5f} max={np.max(y1[:, 1]):.5f}')

    print('')

    # y1_max0 = np.max(y1[:, 0])
    # print(y1_max0)
    # idx = 0
    # last = 0
    # lasts = []
    # for i, item in enumerate(t):
    #     if abs(y1[i][0] - y1_max0) < 23:
    #         idx += 1
    #         print(f'\\par{idx} {item:.1f} {y1[i][0]:.5f}\\\\')
```

```
    #           # print(item - last)
    #           lasts.append(item - last)
    #           last = item

    # lasts.pop(0)
    # print(lasts)
    # print(f'T = {sum(lasts) / len(lasts)}')
    # (11.4 + 11.4 + 11.5 + 11.4) / 4
    # 11.42

    # print('')

    # x1_max0 = np.max(y1[:, 1])
    # print(x1_max0)
    # idx = 0
    # lasts = []
    # for i, item in enumerate(t):
    #     if abs(y1[i][1] - x1_max0) < 0.1:
    #           idx += 1
    #           print(f'\\par{idx} {item:.1f} {y1[i][1]:.5f}\\\\')
    #           # print(item - last)
    #           lasts.append(item - last)
    #           last = item

    # lasts.pop(0)
    # print(lasts)
    # print(f'T = {sum(lasts) / len(lasts)}')

    11.4 + 11.5 + 11.4 + 11.5

    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')
```

```
    ax1.plot(t, y1[:, 0], linewidth=0.5)
    ax2.plot(t, y1[:, 1], linewidth=0.5)

    ax3.set_title('              ␣                    ')
    ax3.set_xlabel('x')
    ax3.set_ylabel('y')
    ax3.plot(y1[:, 0], y1[:, 1], label='3', linewidth=0.5)
    ax3.plot(y2[:, 0], y2[:, 1], label='3', linewidth=0.5)
    ax3.plot(y3[:, 0], y3[:, 1], label='3', linewidth=0.5)
    plt.show()


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