НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

Звіт
з лабораторної роботи № 3
із дисципліни «Методи обчислень-1»
на тему
Наближене розв'язання задачі Коші
для звичайних диференціальних рівнянь та їх систем

Варіант 11

Виконав:

Студент групи КМ-93
Костенко Олександр

Керівник:

викладач Любашенко Н. Д.

Київ-2021

Зміст

Завдання 1

Постановка задачі: розв'язати задачу Коші двома методами:

- Ейлера
- Рунге-Кутта

Початкові умови:

$$f' = \frac{Ax^2 - 1}{y^3 + x^2 + 0.3} = \frac{0.4x^2 - 1}{y^3 + x^2 + 0.3}$$

$$y_0 = -2$$

$$h = 0.2$$

$$\text{Інтервал } [0; 2]$$

Розв'язок:

Метод Ейлера:

$$\begin{cases} y' = \dfrac{0.4x^2 - 1}{y^3 + x^2 + 0.3} \\ y(0) = -2 \end{cases}$$

Відповідно до початкових умов, згідно ітераційних співвідношень при $f(x,y) = \frac{0.4x^2-1}{y^3+x^2+0.3}$ отримуємо значення розв'язку задачі Коші в точках $x_i = x_0 + ih \quad (i = \overline{1,11})$:

$$y_1 = y_0 + h * f(x_0, y_0) = -2 + 0.2\left(\frac{0.4 * 0 - 1}{-8 + 0 + 0.3}\right) = -1.974025974025974$$

$$y_2 = y_1 + h * f(x_1, y_1) = -1.974025974025974$$

$$y_3 = y_2 + h * f(x_2, y_2) = -1.9472589930751643$$

$$y_4 = y_3 + h * f(x_3, y_3) = -1.9202212346947622$$

$$y_5 = y_4 + h * f(x_4, y_4) = -1.893555959 2770825$$

$$y_6 = y_5 + h * f(x_5, y_5) = -1.868117659194371$$

$$y_7 = y_6 + h * f(x_6, y_6) = -1.845126844667507$$

$$y_8 = y_7 + h * f(x_7, y_7) = -1.8264555107129365$$

$$y_9 = y_8 + h * f(x_8, y_8) = -1.8151848052178021$$

$$y_{10} = y_9 + h * f(x_9, y_9) = -1.8167228502364805$$

$$y_{11} = y_{10} + h * f(x_{10}, y_{10}) = -1.840826486972444$$

Результати обчислень заносимо в таблицю:

x_i	0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
$f(x_i)$	-2	-1.974	-1.9473	-1.92	-1.894	-1.868	-1.845	-1.826	-1.815	-1.817	-1.841

Для розрахунків було використано програмний код (дивіться в *Додатку 1*).

Метод Рунге-Кутта:

Тепер для обчислення вихідної функції використаємо метод Рунге-Кутта. Тоді:

$$\begin{cases} y_{i+1} = y_i + \frac{1}{6}(k_1^i + 4k_2^i + k_3^i) \\ k_1^i = f(x_i, y_i) * h \\ k_2^i = f\left(x_i + \frac{h}{2}, y_i + \frac{k_1^i}{2}\right) * h \\ k_3^i = f\left(x_i + h, y_i + 2 * k_2^i - k_1^i\right) * h \end{cases}$$

Згідно зазначених вище формул, при $x_i = x_0 + ih \quad (i = \overline{1,11})$, обчислюємо значення

$f(x_i)$ та заносимо отримане в таблицю:

x_i	0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
$f(x_i)$	-2	*-1.9736*	-1.9466	-1.919	-1.8935	-1.869	-1.848	-1.833	-1.828	-1.839	-1.875

Порівнюючи отримані значення зі значеннями отриманими при застосуванні методу Ейлера, можемо побачити, що вони відрізняються, але незначно.

Для розрахунків було використано програмний код (дивіться в *Додатку 1*).

Завдання 2

Постановка задачі: застосувати метод Рунге-Кутта 4-го порядку для розв'язку системи диференційних рівнянь Лотки-Вольтерра:

$$\begin{cases} \dfrac{dx}{dt} = ax - bxy \\ \dfrac{dy}{dt} = -cy + dxy \end{cases}$$

Яка відповідає заданим початковим умовам:

$$y(t_0) = y_0; x(t_0) = x_0$$

Початкові умови:

$a = 1{,}65;$
$b = 2{,}01;$
$c = 1{,}25;$
$d = 0{,}71$

$$y_0 = 1.92; \quad x_0 = 2.92$$

$$t \in [11;\ 41]$$

Розв'язок:

Початковий крок визначаємо з двох умов:

- $h \le \sqrt[4]{\varepsilon} => h \le 0.177828$
- кількість інтервалів – парне число

Нехай крок $h = 0.15$, тоді кількість інтервалів дорівнює (41-11)/0.15=200 – парне число.

Запишемо робочі формули методу Рунге-Кутта четвертого порядку:

$$t_{i+1} = t_i + h$$

$$y_{i+1} = y_i + \frac{1}{6}(k_1 \ + 2k_2 \ + 2k_3 \ + k_4 \ )$$

$$x_{i+1} = x_i + \frac{1}{6}(q_1 \ + 2q_2 \ + 2q_3 \ + q_4 \ )$$

де

$$k_1 = hf_1(t_i, y_i, x_i)$$

$$q_1 = hf_2(t_i, y_i, x_i)$$

$$k_2 = hf_1(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}, x_i + \frac{q_1}{2})$$

$$q_2 = hf_2\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}, x_i + \frac{q_1}{2}\right)$$

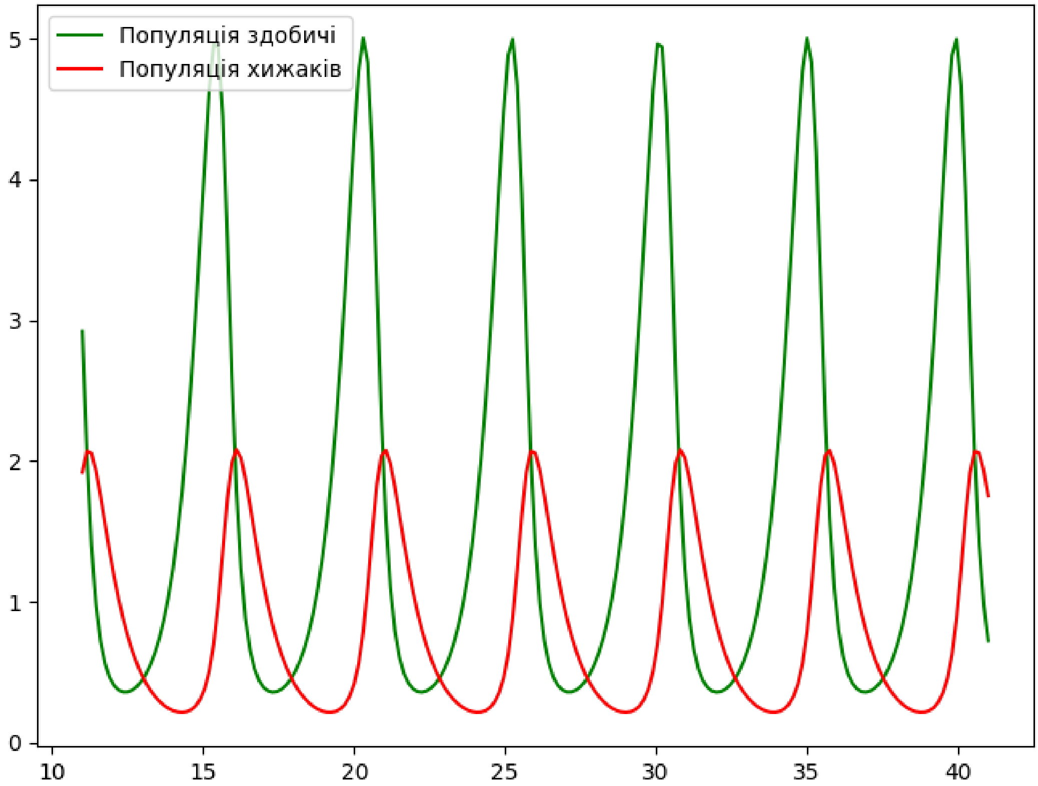
$$k_3 = hf_1\left(t_i + \frac{h}{2}, y_i + \frac{k_2}{2}, x_i + \frac{q_2}{2}\right)$$

$$q_3 = hf_2(t_i + \frac{h}{2}, y_i + \frac{k_2}{2}, x_i + \frac{q_2}{2})$$

$$k_4 = hf_1(t_i + h, y_i + k_3, x_i + q_3)$$

$$q_4 = hf_2(t_i + h, y_i + k_3, x_i + q_3)$$

Використовуючи програмні засоби, зобразимо залежності $y(t\ \ )$ та $x(t\ \ )$, де $x(t\ \ )$ – популяція здобичі та $y(t\ \ )$ – популяція хижаків:

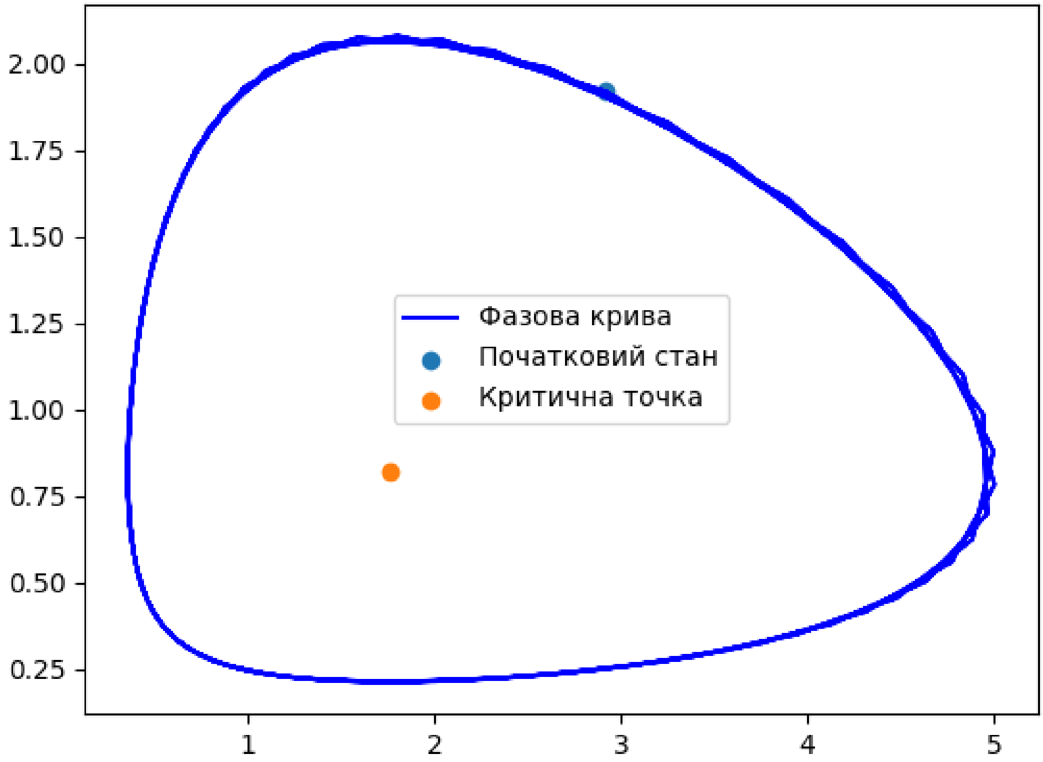


Використовуючи програмні засоби, зобразимо залежність $y(x)$, що є фазовою кривою, початковий стан (2.92; 1.92) та критичну точку:

Критична точка:

$$\begin{cases} \frac{dx}{dt} = 0 \\ \frac{dy}{dt} = 0 \end{cases} => \begin{cases} ax - bxy = 0 \\ -cy + dxy = 0 \end{cases} => \begin{cases} x(a - by) = 0 \\ y(-c + dx) = 0 \end{cases} => \left[\begin{array}{l} \begin{cases} x = 0 \\ y = 0 \end{cases} \\ \begin{cases} x = \frac{c}{d} \\ y = \frac{a}{b} \end{cases} \end{array} \right.$$

Стаціонарний стан з координатами (0;0) є тривіальним і не представляє інтересу (оскільки якщо кількість жертв = 0 та хижаків = 0, то нічого не відбувається). Друга критична точка $\left(\frac{c}{d}; \frac{a}{b}\right)$ (в нашому випадку $\left(\frac{1.25}{0.71}; \frac{1.65}{2.01}\right)$) є більш цікавою. Зобразимо її на графіку. В цій точці похідна дорівнює нулю, а отже популяція не буде змінюватися.



Код програмного засобу дивіться в Додатку 2.

Додаток 1

```
#Task 1
#1
y0=-2
a=0.4
h=0.2

x=[]
y=[]
for i in range(0,21,2):
    x.append(i/10)
    y.append(0)

y[0]=y0

def f(x, y):
    f=(a*x**2-1)/(y**3+x**2+0.3)
    return f


for i in range(len(y)-1):
    y[i+1]=y[i]+h*f(x[i], y[i])

print(x)
print(y)

#2
y=[]
for i in range(0,21,2):
    y.append(0)

y[0]=y0
k1=[]
k2=[]
k3=[]

for i in range(len(y)-1):
    k1.append(f(x[i],y[i])*h)
    k2.append(f(x[i]+h/2, y[i]+k1[i]/2)*h)
    k3.append(f(x[i]+h, y[i]+2*k2[i]-k1[i])*h)
    y[i+1]=y[i]+(1/6)*(k1[i]+4*k2[i]+k3[i])

print(y)
```

Додаток 2

```
from matplotlib import pyplot as plt

#Task2
a=1.65
b=2.01
c=1.25
d=0.71

x0=2.92
y0=1.92

e=10**(-3)

h=0.15
t=[11]
y=[y0]
x=[x0]

k1=[]
k2=[]
k3=[]
k4=[]

q1=[]
q2=[]
q3=[]
q4=[]

def fx(t,x,y):
    return (a*x-b*x*y)

def fy(t,x,y):
    return (-c*y+d*x*y)

for i in range(int(30/h)):
    t.append(0)
    x.append(0)
    y.append(0)

for i in range(int(30/h)):

    t[i+1]=round(t[i]+h,2)

    k1.append(h*fy(t[i],x[i],y[i]))
```

```
    q1.append(h*fx(t[i], x[i], y[i]))

    k2.append(h*fy(t[i]+h/2, x[i]+q1[i]/2, y[i]+k1[i]/2))
    q2.append(h*fx(t[i]+h/2, x[i]+q1[i]/2, y[i]+k1[i]/2))

    k3.append(h*fy(t[i]+h/2, x[i]+q2[i]/2, y[i]+k2[i]/2))
    q3.append(h*fx(t[i]+h/2, x[i]+q2[i]/2, y[i]+k2[i]/2))

    k4.append(h*fy(t[i]+h, x[i]+q3[i], y[i]+k3[i]))
    q4.append(h*fx(t[i]+h, x[i]+q3[i], y[i]+k3[i]))

    y[i+1]=y[i]+(1/6)*(k1[i]+2*k2[i]+2*k3[i]+k4[i])
    x[i+1]=x[i]+(1/6)*(q1[i]+2*q2[i]+2*q3[i]+q4[i])


fig, ax=plt.subplots(figsize=(8,6))
ax=plt.plot(t, x, color="green", label="Популяція здобичі")
ax=plt.plot(t, y, color="red", label="Популяція хижаків")

plt.legend()
plt.show()

ax=plt.plot(x,y, color="blue", label="Фазова крива")
ax=plt.scatter(x[0], y[0], label="Початковий стан")
ax=plt.scatter(c/d, a/b, label="Критична точка")

plt.legend()
plt.show()
```