Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Казанский (Приволжский) Федеральный университет

ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
Кафедра прикладной математики и искусственного интеллекта

Направление подготовки: 01.03.04 - «Прикладная математика»

КУРСОВАЯ РАБОТА

по дисциплине: «Численные методы»

Система типа хищник-жертва. Модель Мак-Артура.

Студент 2 курса
группы 09-222
«____» ________ 2024 г. ______________ Л.Ф. Фаррахова

Научный руководитель
ассистент б. с.
«____» ________ 2024 г. ______________ О.В. Глазырина

Казань, 2024

Содержание

1 Постановка задачи

Рассмотрим динамику популяции двух видов, взаимодействующих между собой по типу хищник-жертва, при наличии внутривидовой конкуренции жертв за ограниченные ресурсы и при учете фактора насыщения хищника. Обозначим через $x = x(t)$ и $y = y(t)$ плотности популяций жертв и хищников в момент времени t. Уравнения имеют следующий вид:

$$\begin{aligned} \frac{dx}{dt} &= a\left(1 - \frac{x}{K}\right)x - \frac{bxy}{1+Ax}, \\ \frac{dy}{dt} &= -cy + \frac{d}{1+Ax}xy. \end{aligned} \tag{1}$$

где a,b,c,d,A,K - неотрицательные числа. Структура уравнений следующая:

- Слагаемое $a\left(1 - \frac{x}{K}\right)x$ определяет скорость размножения жертв в отсутствии хищников. При малых x скорость определяется величиной a и, таким образом, а характеризует норму рождаемости при малой плотности популяции. При большой плотности (до величины K) популяция растёт, при $x > K$ - уменьшается в размерах (скорость отрицательна). Таким образом, это слагаемое описывает ограниченность ресурсов: окружающая среда может обеспечивать существование только популяции плотности меньшей K.

- Слагаемое $\frac{bxy}{1+Ax}$ описывает влияние хищников на популяцию жертв. Функция $\frac{bx}{1+Ax}$ характеризует количество жертв, убиваемых одним хищником в единицу времени (реакция хищника на плотность популяции жертвы). Здесь учтено, что при большой плотности жертв хищник убивает лишь $\frac{b}{A}$ жертв в единицу времени; то есть перестаёт убивать когда насыщается.

- Второе уравнение определяет изменение популяции хищников. Постоянная c определяется естественной нормой смертности хищников. Второе слагаемое характеризует прирост хищников в зависимости от плотности жертв (функция $\dfrac{d}{1+Ax}$). При большой плотности жертв скорость прироста хищников определяется величиной $\dfrac{d}{A}$ и, таким образом, $\dfrac{d}{A}$ характеризует норму рождаемости при благоприятных для хищников условиях.

Безразмерная форма уравнений

Вводя безразмерные величины

$$X = \left(\frac{d}{c}\right) x, \quad Y = \left(\frac{\mathrm{b}}{\mathrm{a}}\right) \mathrm{y}, \quad \tau = \mathrm{at}, \quad \alpha = \frac{\mathrm{Ac}}{\mathrm{d}}, \quad \epsilon = \frac{\mathrm{c}}{\mathrm{Kd}}, \quad \gamma = \frac{\mathrm{c}}{\mathrm{a}},$$

преобразуем уравнения (1) к виду

$$\begin{aligned} \frac{dX}{d\tau} &= (1-\epsilon X)X - \frac{XY}{1+\alpha X}, \\ \frac{dY}{d\tau} &= \gamma\left(\frac{X}{1+\alpha X} - 1\right)Y, \end{aligned} \tag{2}$$

И дополним их начальньми условиями

$$\mathbf{X}(0) = \mathbf{X}_0, \mathbf{Y}(0) = \mathbf{Y}_0 \tag{3}$$

Метод решения задачи Коши

Для решения задачи (2) - (3) использовать метод Рунге-Кутты 4-го порядка точности (правило 3/8):

$$\begin{aligned} k_1 &= f\left(t_n, y_n\right), \\ k_2 &= f\left(t_n + h/3, y_n + hk_1/3\right), \\ k_3 &= f\left(t_n + 2h/3, y_n - hk_1/3 + hk_2\right), \\ k_4 &= f\left(t_n + h, y_n + hk_1 - hk_2 + hk_3\right), \\ y_{n+1} &= y_n + h\left(k_1 + 3k_2 + 3k_3 + k_4\right)/8; \end{aligned} \tag{4}$$

с постояннмь шагом h. Для проверки правильности работы программы решить тестовую задачу из двух уравнений

$$y_1' = y_1/(2+2t) - 2\mathrm{ty}_2, \quad y_2' = y_2/(2+2t) + 2ty_1 \tag{5}$$

на отрезке $[0,2]$ с точным решением

$$y_1 = \cos\left(t^2\right)\sqrt{1+t}, \quad y_2 = \sin\left(t^2\right)\sqrt{1+t}$$

В ходе работы необходимо выполнить следующие действия:

1. Проверить правильность вывода исходных уравнений (1), уравнений в безразмерном виде (2) и тестового решения (5).

2. Найти стационарные решения (состояния равновесия) системы (2)

3. Написать процедуру интегрирования задачи Коши для системы из n обыкновенных дифференциальных уравнений по формулам (4) на произвольном отрезке $[a,b]$ с постоянным шагом h.

4. Для тестовой задачи (5) построить графики зависимости максимальной погрешности решения e и e/h^4 от выбранного шага h.

5. Для значений параметра α из интервала $[0.1, 0.9]$ рассчитать динамику популяции при следующих исходных данных

$$\epsilon = 0.1, \quad \gamma = 1, \quad X_0 = 3, \quad Y_0 = 1$$

   Определите те значения параметра α при которых в системе появляются и исчезают устойчивые колебания (предельный цикл). Приведите графики наиболее характерных решений в координатах (X,Y), $(\mathrm{X(t),t})$ и $(\mathrm{Y(t),t})$ и дайте их интергретацию.

3 Ход работы

Для перехода от системы уравнений (1) к системе уравнений (2) введем новые переменные и новые параметры, а затем проведем соответствующие замены и преобразования.

Введем следующие замены:

$$X = \frac{d}{c}x,$$
$$Y = \frac{b}{a}y,$$
$$\tau = at,$$

где τ - новая безразмерная переменная времени.

Также введем новые параметры:

$$\alpha = \frac{Ac}{d},$$
$$\epsilon = \frac{c}{Kd},$$
$$\gamma = \frac{c}{a}.$$

Найдем производные:

$$\frac{dx}{dt} = \frac{d\left(\frac{c}{d}X\right)}{d\left(\frac{\tau}{a}\right)} = \frac{c}{d}\cdot a\frac{dX}{d\tau}, \quad \frac{dy}{dt} = \frac{d\left(\frac{a}{b}Y\right)}{d\left(\frac{\tau}{a}\right)} = \frac{a}{b}\cdot a\frac{dY}{d\tau}.$$

Подставим новые переменные и производные в исходные уравнения:

$$\frac{c}{d}\cdot a\frac{dX}{d\tau} = a\left(1 - \frac{\frac{c}{d}X}{K}\right)\frac{c}{d}X - \frac{b\frac{c}{d}X\frac{a}{b}Y}{1 + A\frac{c}{d}X},$$

$$\frac{a}{b}\cdot a\frac{dY}{d\tau} = -c\cdot\frac{a}{b}Y + \frac{d\cdot\frac{a}{b}Y\cdot\frac{c}{d}X}{1 + A\frac{c}{d}X}.$$

Сократим на соответствующие коэффициенты:

$$\frac{ca}{d}\frac{dX}{d\tau} = a\left(1 - \frac{cX}{Kd}\right)\frac{cX}{d} - \frac{cXY}{d(1+\dfrac{AcX}{d})},$$

$$a\frac{dY}{d\tau} = -c\frac{aY}{b} + \frac{aXY}{b(1+\dfrac{AcX}{d})}.$$

Применив параметры $\epsilon = \dfrac{c}{Kd}$, $\alpha = \dfrac{Ac}{d}$ и $\gamma = \dfrac{c}{a}$, получим систему уравнений (2):

$$\frac{dX}{d\tau} = (1 - \epsilon X)X - \frac{XY}{1+\alpha X},$$
$$\frac{dY}{d\tau} = \gamma\left(\frac{X}{1+\alpha X} - 1\right)Y.$$

дополненную начальными условиями:

$$X(0) = X_0, \quad Y(0) = Y_0.$$

Для того, чтобы проверить правильность работы метода Рунге-Кутты, решим тестовую задачу (5).

Для доказательства, что $y_1(t) = \cos(t^2)\sqrt{1+t}$ и $y_2(t) = \sin(t^2)\sqrt{1+t}$ являются решениями данной системы дифференциальных уравнений, мы должны подставить их в уравнения и проверить, что они удовлетворяют исходной системе.

Подставим $y_1(t) = \cos(t^2)\sqrt{1+t}$ и $y_2(t) = \sin(t^2)\sqrt{1+t}$ в первое уравнение:

$$y_1' = \frac{y_1}{2+2t} - 2ty_2 = \frac{\cos(t^2)\sqrt{1+t}}{2+2t} - 2t\sin(t^2)\sqrt{1+t}.$$

Подставим $y_1(t) = \cos(t^2)\sqrt{1+t}$ и $y_2(t) = \sin(t^2)\sqrt{1+t}$ во второе уравнение:

$$y_2' = \frac{y_2}{2+2t} + 2ty_1 = \frac{\sin(t^2)\sqrt{1+t}}{2+2t} + 2t\cos(t^2)\sqrt{1+t}.$$

Теперь проверим, удовлетворяют ли эти выражения исходной системе уравнений:

$$
\begin{aligned}
\frac{\cos(t^2)\sqrt{1+t}}{2+2t} - 2t\sin(t^2)\sqrt{1+t} &= \frac{y_1}{2+2t} - 2ty_2 = y_1',\\
\frac{\sin(t^2)\sqrt{1+t}}{2+2t} + 2t\cos(t^2)\sqrt{1+t} &= \frac{y_2}{2+2t} + 2ty_1 = y_2'.
\end{aligned}
$$

Таким образом, подставив $y_1(t) = \cos(t^2)\sqrt{1+t}$ и $y_2(t) = \sin(t^2)\sqrt{1+t}$ в исходную систему дифференциальных уравнений, мы видим, что они являются решениями этой системы.

Построим график полученных решений методом Рунге-Кутты и график точного решения для погрешностей для $x(t)$ и $y(t)$, выбрав количество разбиений равным 100.

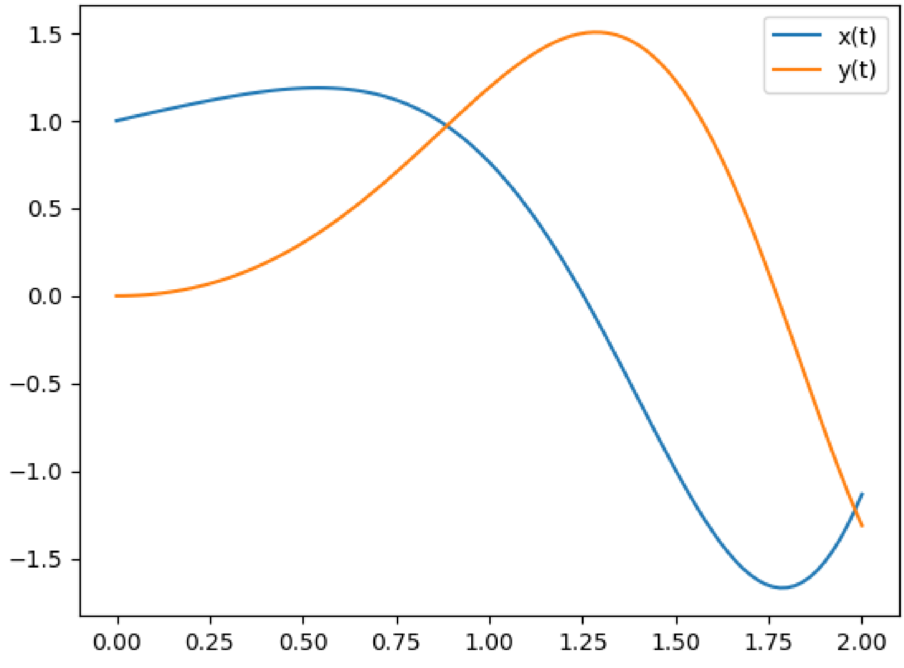

Рисунок 1 - График полученного решения методом Рунге-Кутты

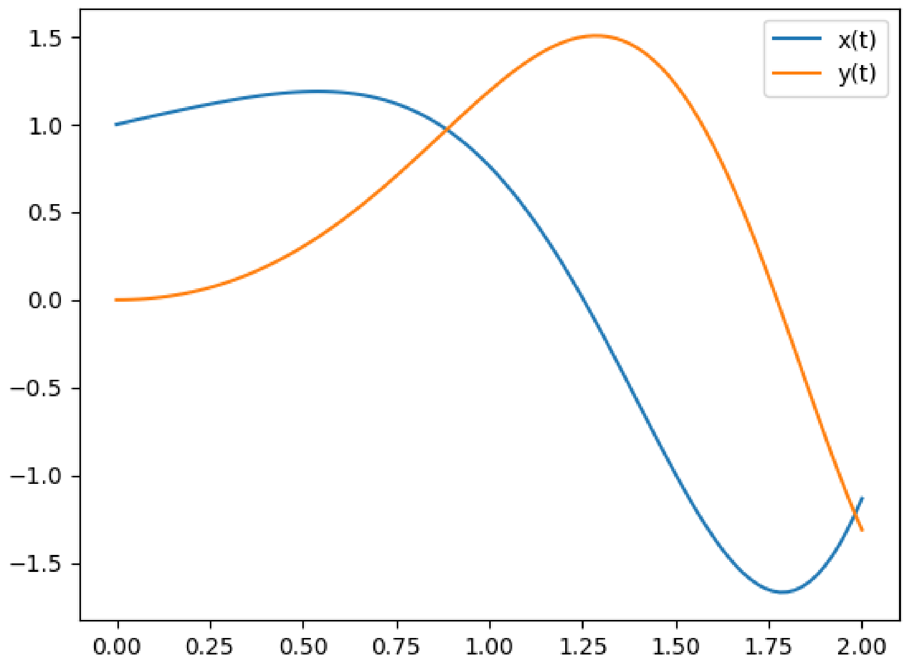

Рисунок 2 - График точного решения

Сравнивая рисунки 1-2 визуально определить погрешность достаточно сложно, поэтому вычислим максимальную погрешность e и e/h^4 в зависимости от n и построим график, где по оси ординат - максимальная ошибка, а по оси абсцисс количество разбиений(n):

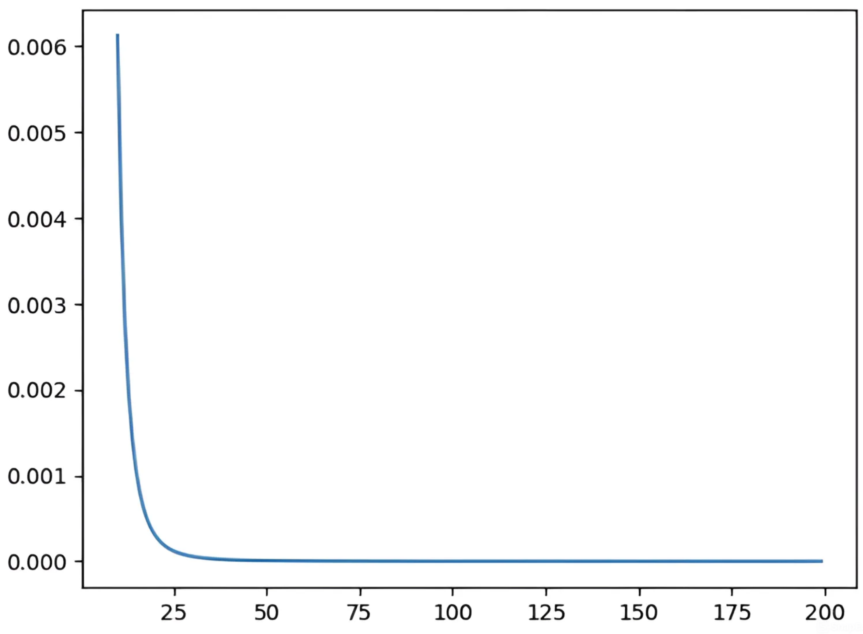

Рисунок 3 - График зависимости погрешности от шага h.

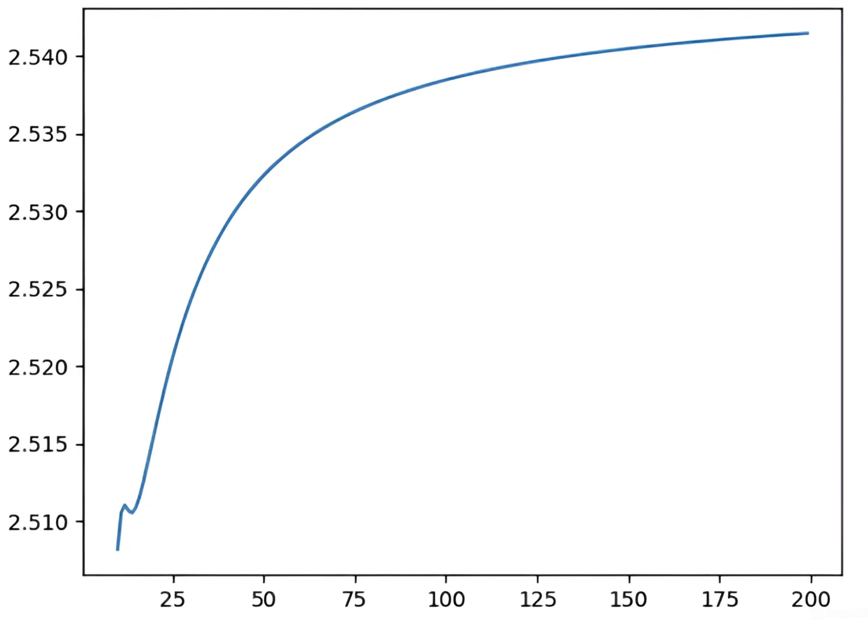

Рисунок 4 - График зависимости погрешности от шага h^4.

По графикам можно сделать вывод, что максимальная погрешность решения е возрастает при увеличении шага h.

Далее найдем стационарные решения системы (2). Для этого приравняем производняе к нулю и решим полученные уравнения. Итак, для данной системы уравнений стационарные решения будут удовлетворять следующим уравнениям:

$$(1-\epsilon X)X - \frac{XY}{1+\alpha X} = 0,$$
$$\gamma\left(\frac{X}{1+\alpha X} - 1\right)Y = 0,$$

Первое уравнение можно переписать в виде:

$$X\left(1-\epsilon X - \frac{Y}{1+\alpha X}\right) = 0$$

Отсюда следует, что

$$X = 0 \text{ или } 1-\epsilon X - \frac{Y}{1+\alpha X} = 0$$

Второе уравнение можно переписать в виде:

$$Y\left(\gamma\left(\frac{X}{1+\alpha X}\right)\right)=0$$

Отсюда следует, что

$$Y=0 \text{ или } \gamma\left(\frac{X}{1+\alpha X}-1\right)=0$$

Таким образом, решив эту систему уравнений, мы найдем состояния равновесия системы дифференциальных уравнений, которые имеют вид:

$$(X,Y)=(0,0) \text{ или } (X,Y)=(-\frac{1}{\alpha},\gamma\frac{1-\alpha}{\alpha})$$

Далее для значений параметра α из интервала $[0.1, 0.9]$ рассчитаем динамику популяции при следующих исходных данных

$$\epsilon=0.1, \quad \gamma=1, \quad X_0=3, \quad Y_0=1$$

Приведем графики в координатах $(X,Y),(X(t),t)$ и $(Y(t),t)$

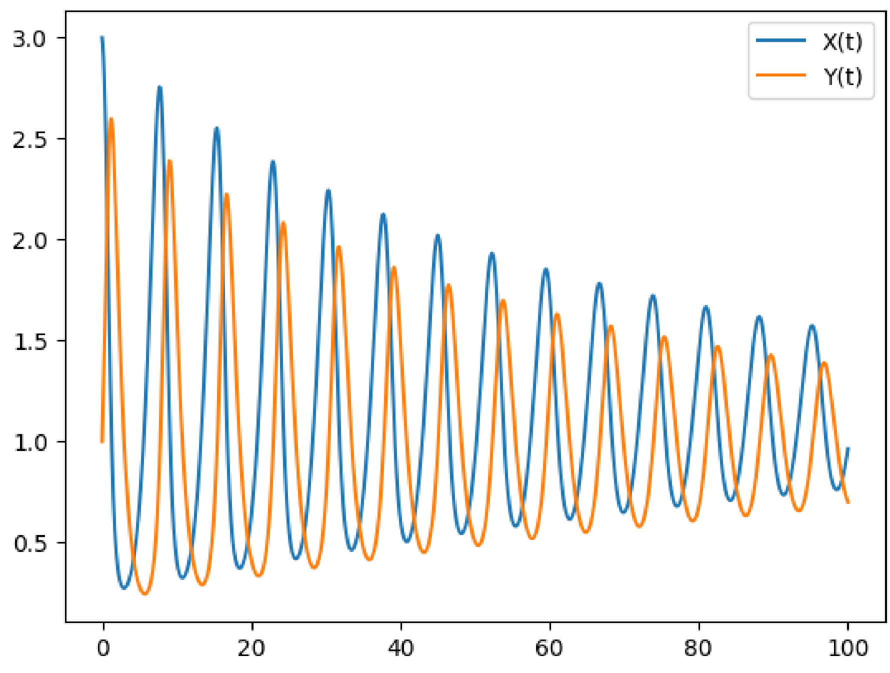

Рисунок 5 - График динамики популяции при $\alpha=0.1$ в координатах $(X(t),t),(Y(t),t)$

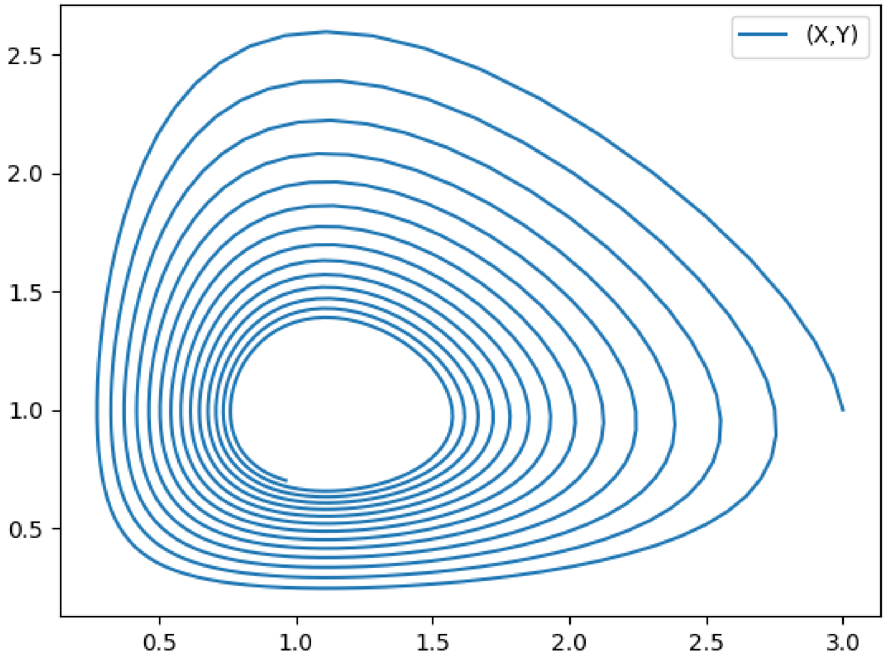

Рисунок 6 - График динамики популяции при $\alpha = 0.1$ в координатах (X, Y)

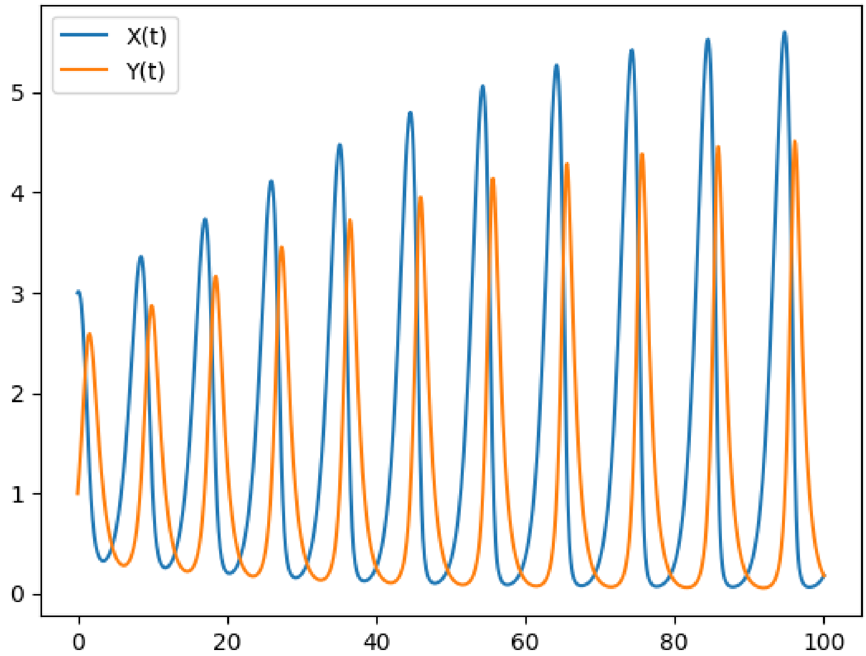

Рисунок 7 - График динамики популяции при $\alpha = 0.2$ в координатах $(X(t), t), (Y(t), t)$

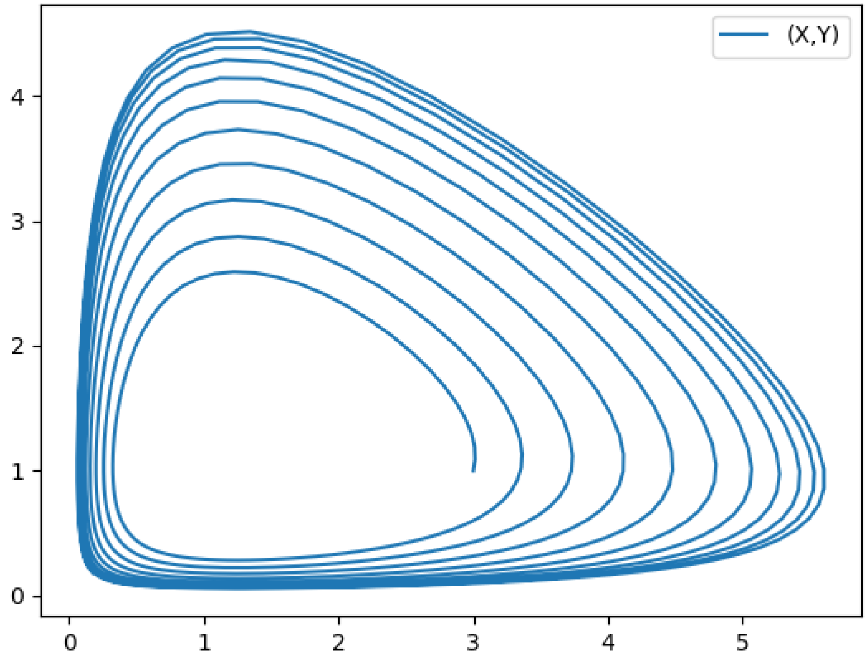

Рисунок 8 - График динамики популяции при $\alpha = 0.2$ в координатах (X, Y)

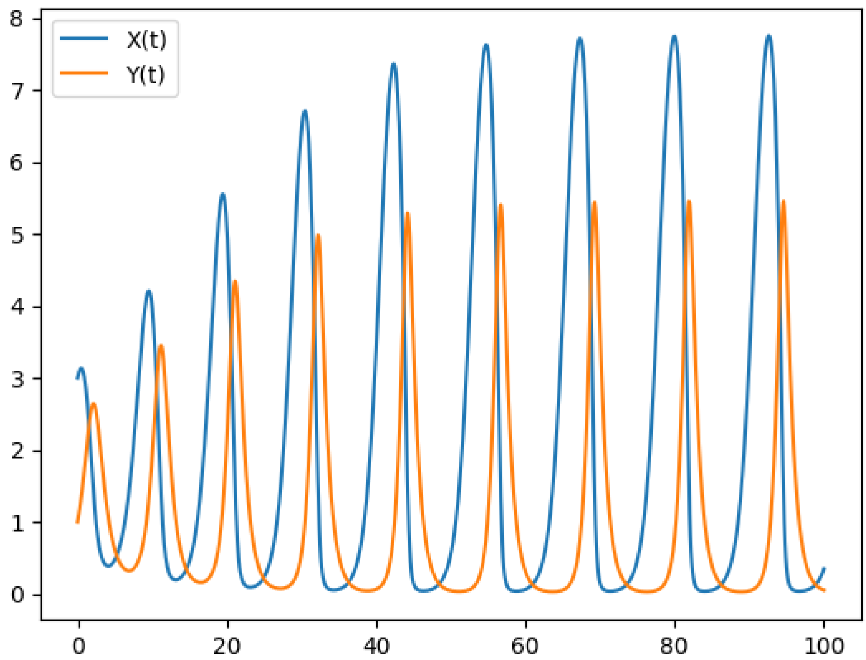

Рисунок 9 - График динамики популяции при $\alpha = 0.3$ в координатах $(X(t), t), (Y(t), t)$

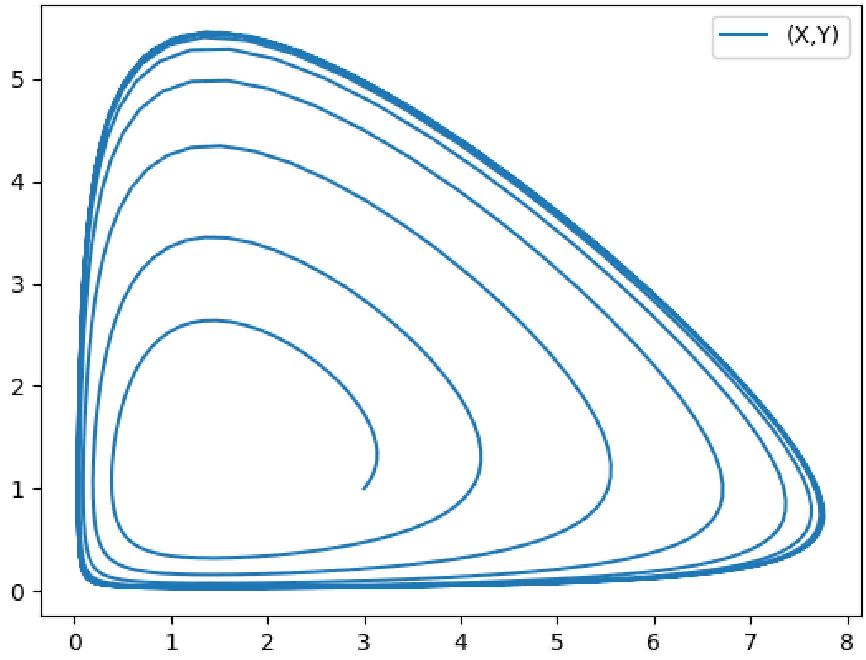

Рисунок 10 - График динамики популяции при $\alpha = 0.3$ в координатах (X, Y)

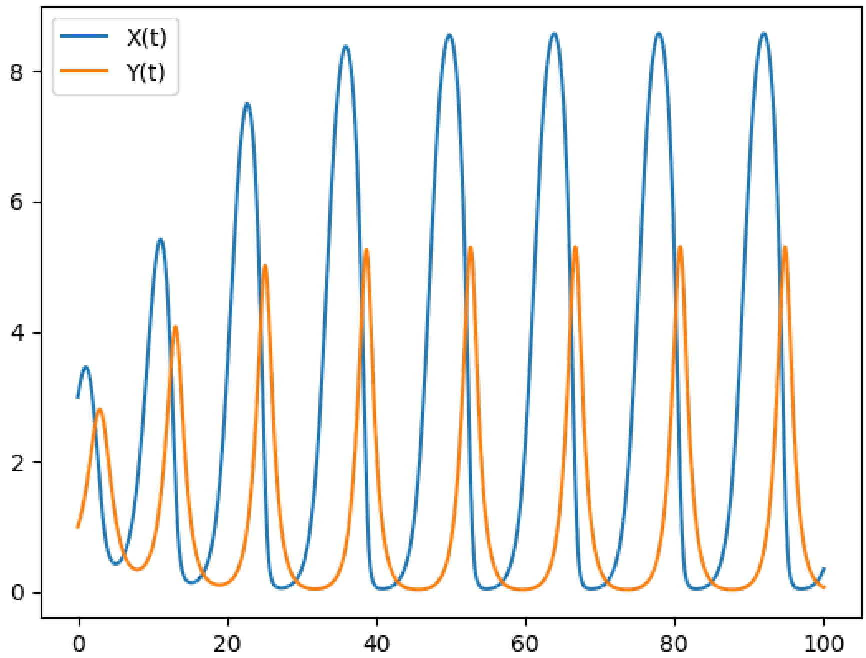

Рисунок 11 - График динамики популяции при $\alpha = 0.4$ в координатах $(X(t), t), (Y(t), t)$

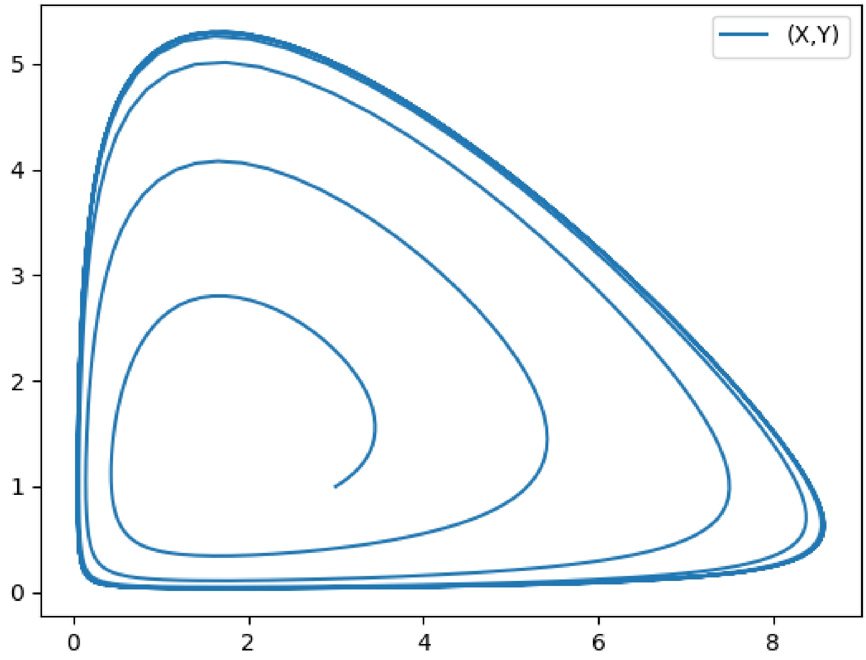

Рисунок 12 - График динамики популяции при $\alpha = 0.4$ в координатах (X, Y)

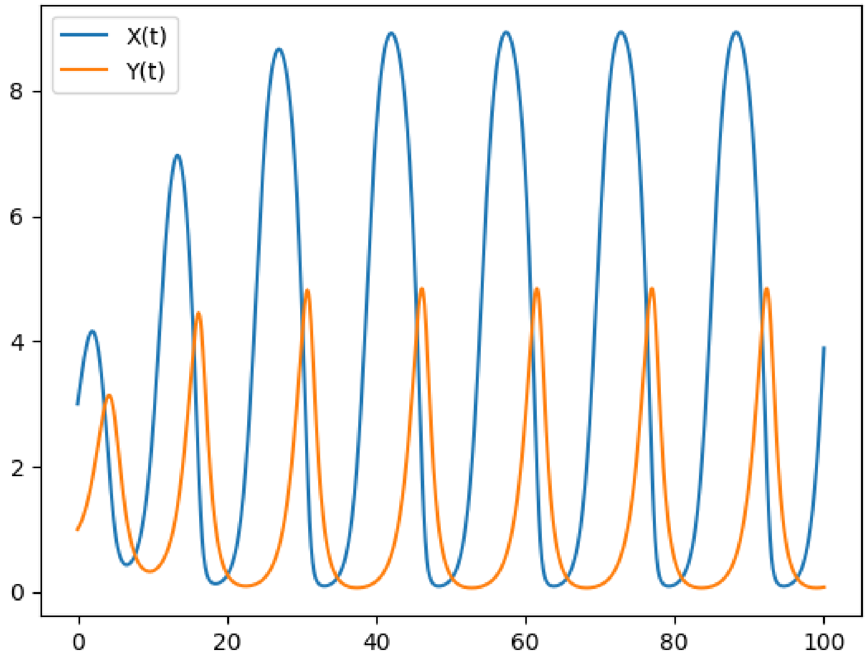

Рисунок 13 - График динамики популяции при $\alpha = 0.5$ в координатах $(X(t), t), (Y(t), t)$

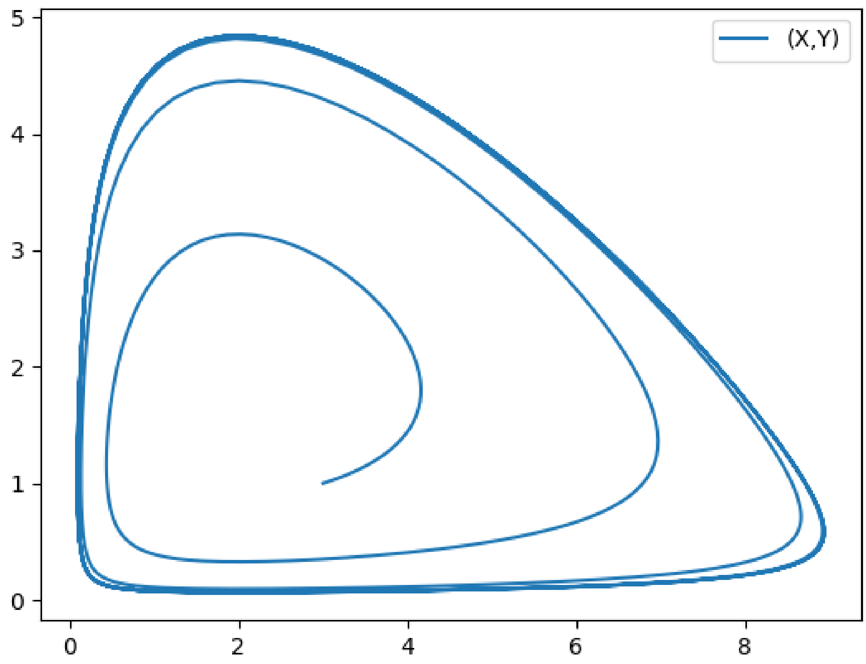

Рисунок 14 - График динамики популяции при $\alpha = 0.5$ в координатах (X, Y)

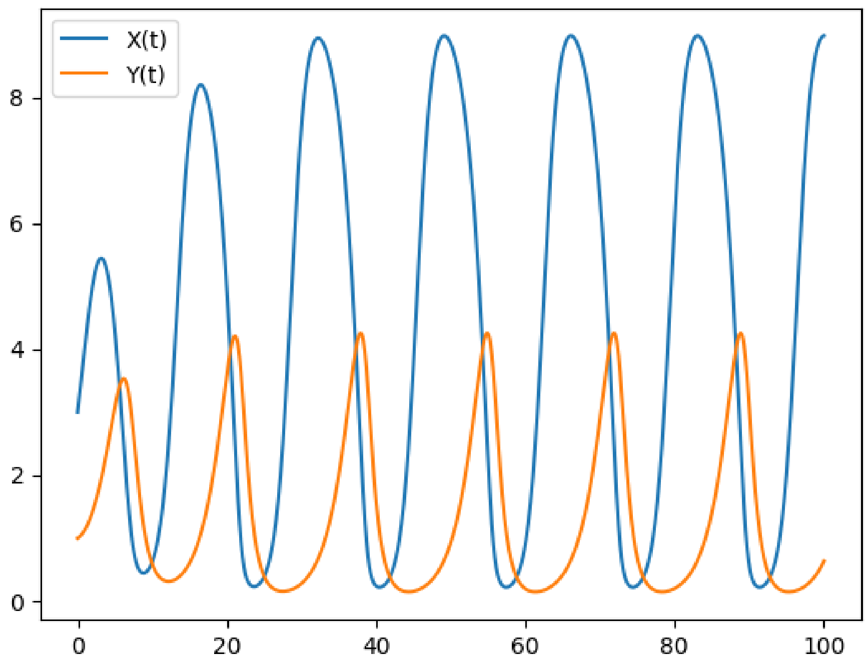

Рисунок 15 - График динамики популяции при $\alpha = 0.6$ в координатах $(X(t), t), (Y(t), t)$

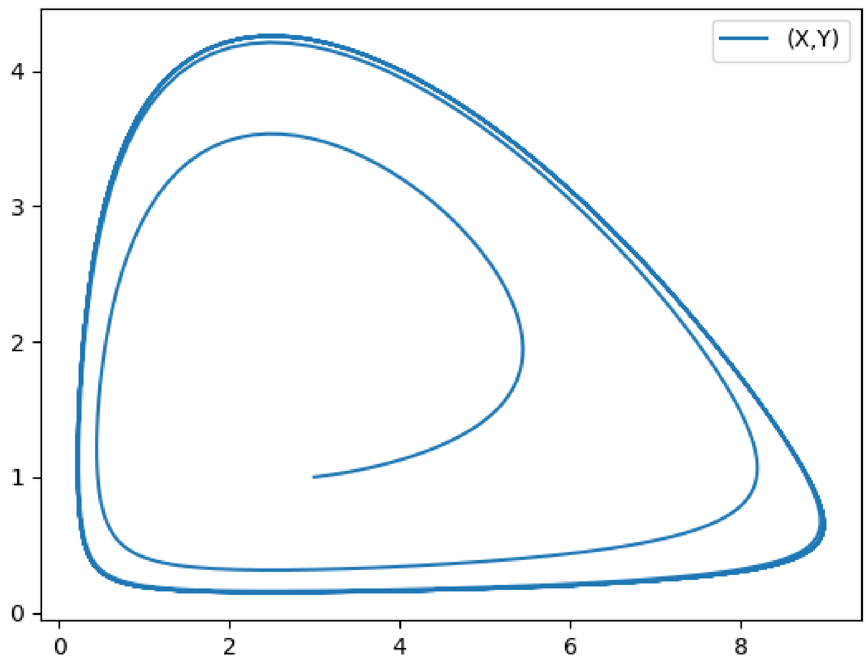

Рисунок 16 - График динамики популяции при $\alpha = 0.6$ в координатах (X, Y)

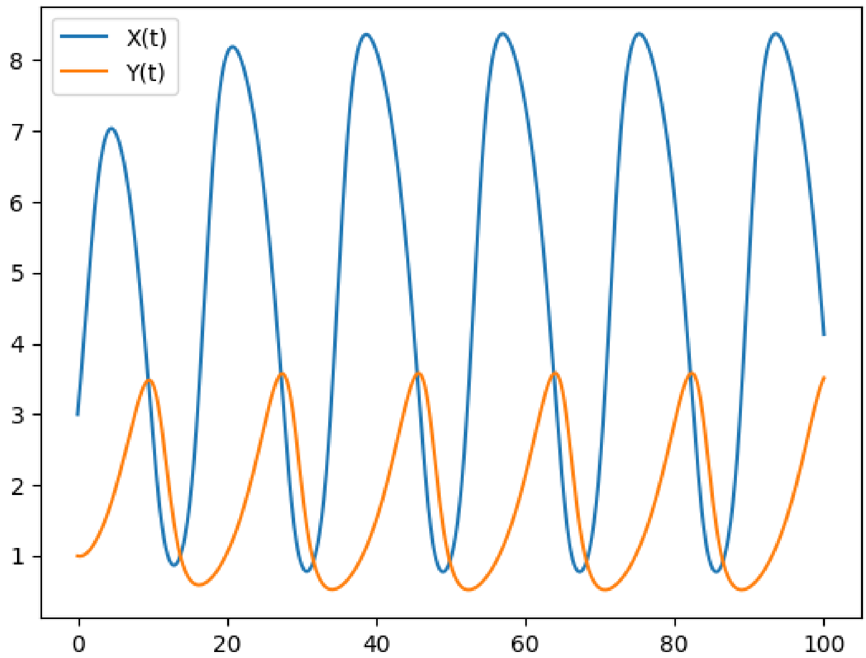

Рисунок 17 - График динамики популяции при $\alpha = 0.7$ в координатах $(X(t), t), (Y(t), t)$

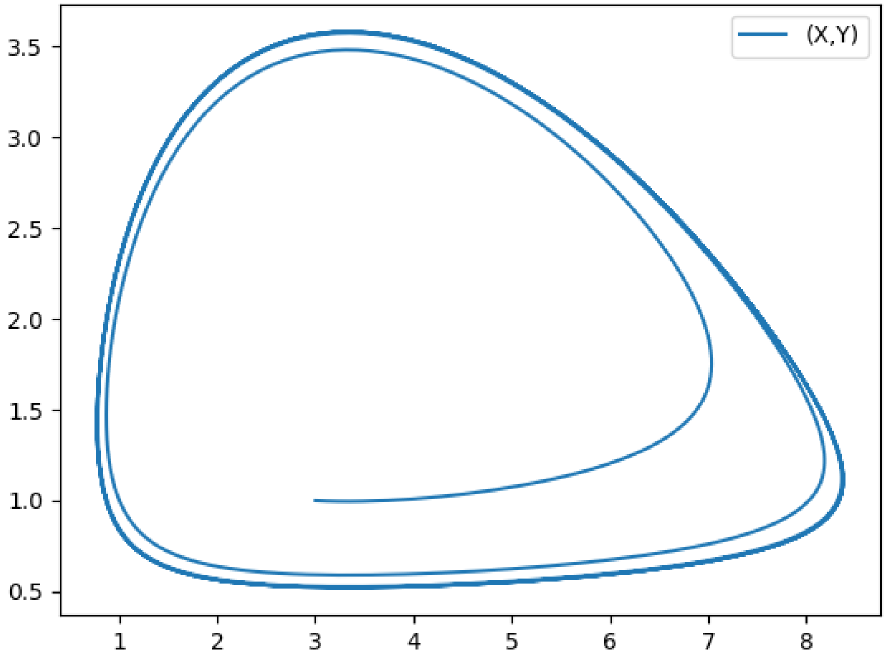

Рисунок 18 - График динамики популяции при $\alpha = 0.7$ в координатах (X, Y)

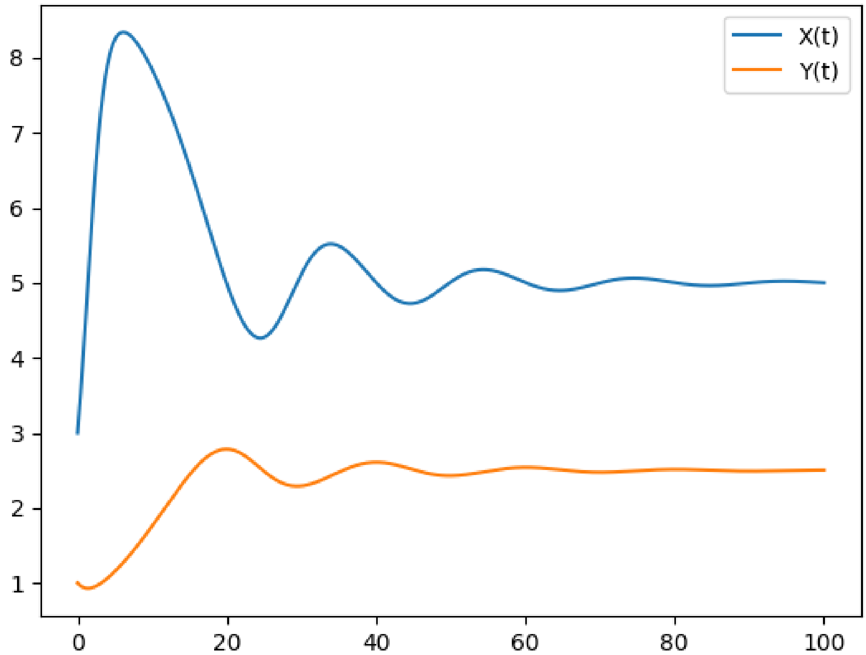

Рисунок 19 - График динамики популяции при $\alpha = 0.8$ в координатах $(X(t), t), (Y(t), t)$

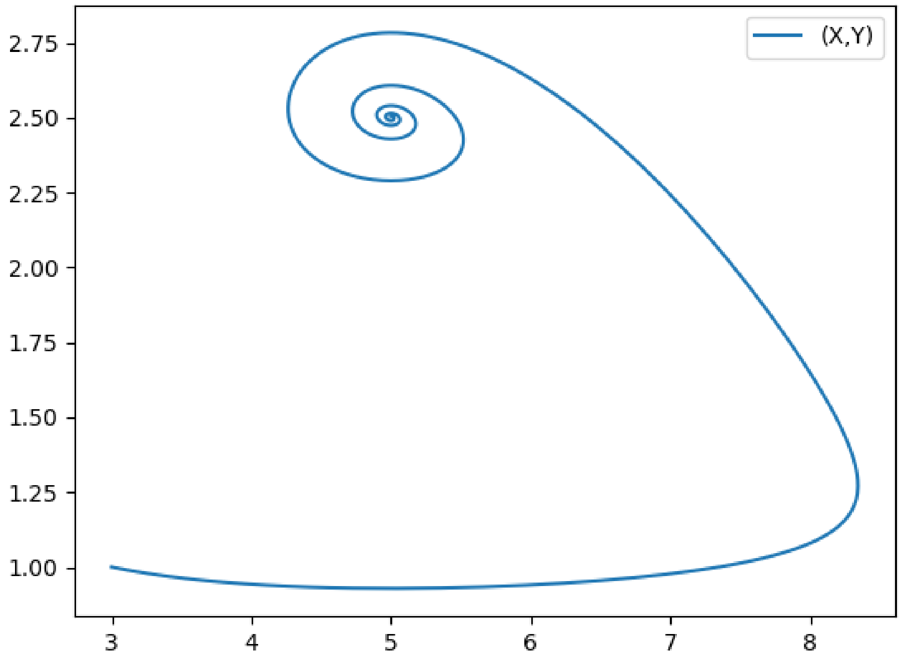

Рисунок 20 - График динамики популяции при $\alpha = 0.8$ в координатах (X, Y)

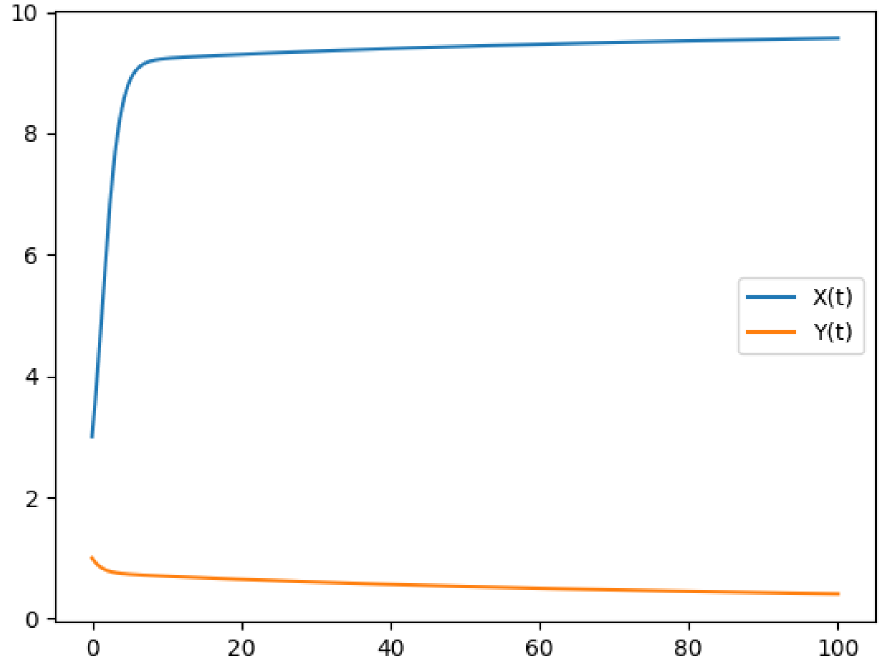

Рисунок 21 - График динамики популяции при $\alpha = 0.9$ в координатах $(X(t), t), (Y(t), t)$

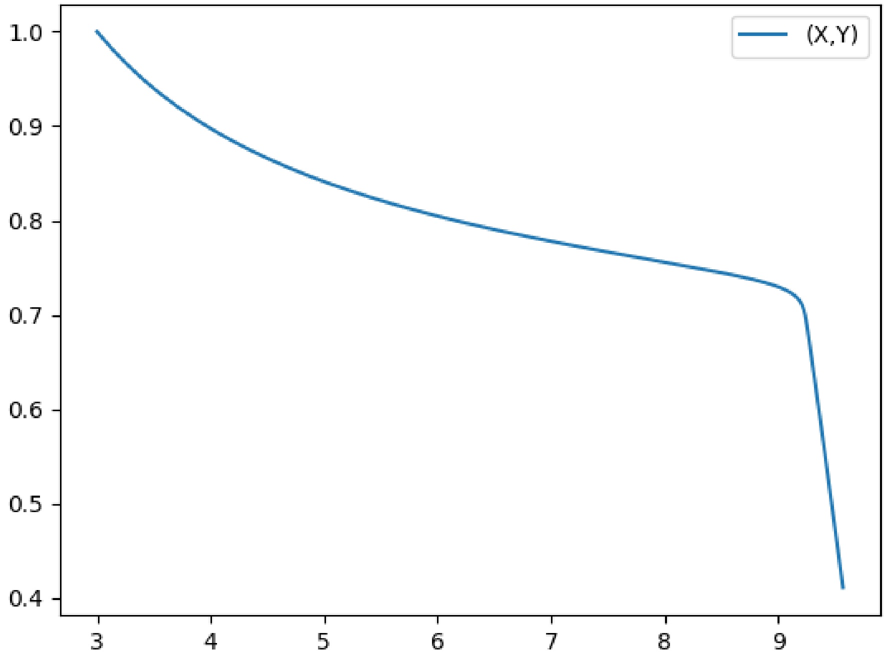

Рисунок 22 - График динамики популяции при $\alpha = 0.9$ в координатах (X, Y)

На графиках в координатах (X, Y) мы можем увидеть, как изменяются значения переменных X и Y в зависимости от параметра α.

При $\alpha = 0.1$ и $\alpha = 0.8$ система имеет устойчивый фокус, что означает, что популяции хищников и жертв со временем достигают равновесия. В этом случае популяции колеблются вокруг точки равновесия и в конечном итоге стабилизируются, не продолжая изменяться со временем.

Устойчивые фокусы указывают на то, что при определенных условиях экосистема может прийти в состояние стабильности, где популяции остаются постоянными.

При $\alpha \in (0.1, 0.7)$ система имеет неустойчивый фокус, что приводит к колебаниям популяций с увеличивающейся амплитудой до тех пор, пока они не достигнут предельного цикла. Предельный цикл представляет собой состояние, при котором популяции колеблются периодически и стабильно без затухания или роста амплитуды.

Неустойчивые фокусы и предельные циклы указывают на динамичную экосистему, где популяции хищников и жертв регулярно меняются, но эти изменения ограничены и предсказуемы.

Так, α влияет на насыщение хищников, т.е. на то, как быстро хищники достигают своего предельного уровня добычи при увеличении плотности популяции жертв.

Низкие значения α (например, 0.1) могут соответствовать ситуации, когда хищники быстро насыщаются и перестают увеличивать уровень добычи, что способствует установлению равновесия. Средние значения α (например, 0.5) приводят к неустойчивым колебаниям и предельным циклам, что указывает на более сложные динамические взаимодействия между хищниками и жертвами. Высокие значения α (например, 0.8) могут указывать на ситуацию, когда хищники медленно достигают насыщения, что также приводит к устойчивому равновесию.

Таким образом, учет насыщения хищников играет ключевую роль в динамике популяций. Он определяет, как хищники реагируют на изменения в популяции жертв и влияет на устойчивость системы. Внутривидовая конкуренция среди жертв также важна для достижения равновесия и предотвращения неконтролируемого роста популяций.

Понимание того, при каких значениях параметров система переходит от устойчивого равновесия к колебаниям, важно для управления экосистемой. Это позволяет предсказать и, возможно, контролировать динамику популяций в реальных условиях.

4 Вывод

В ходе выполнении курсовой работы, была решена тестовая задача и реализована программа интегрирования задачи Коши с помощью Рунге-Кутты 4-го порядка точности с постоянным шагом h. Были построены графики зависимости максимальной погрешности решения e и e/h^4 от выбранного шага h. По графикам можно сделать вывод, что максимальная погрешность решения e возрастает при увеличении шага h. Была построена модель Мак – Артура системы типа хищник-жертва.

5 Список литературы

1. Даутов Р. З. Практикум по дисциплине “Численные методы”. Решение задачи Коши для системы ОДУ. - Учебное пособие, Казань, 2014.

2. Эрроусмит Д., Плейс К. Обыкновенные дифференциальные уравнения. Качественная теория с приложениями. - М.: Мир, 1986.

3. Хайрер Э., Нерсетт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. - М.: Мир, 1990.

4. Самарский А.А., Гулин А.В. Численные методы. - М.: Наука, 1989.

5. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. - М.: Наука, 1987.

6 Листинг

Программа реализована на Python.

```
import numpy as np
import matplotlib.pyplot as plt


def fx_test(t, x, y):
    return x / (2 + 2 * t) - 2 * t * y


def fy_test(t, x, y):
    return y / (2 + 2 * t) + 2 * t * x


def fx_dynamic(t, x, y, alfa):
    eps = 0.1
    return (1 - eps * x) * x - (x * y) / (1 + alfa * x)


def fy_dynamic(t, x, y, alfa):
    gamma = 1
    return gamma * (x / (1 + alfa * x) - 1) * y


def solve_test(t1, t2, h, x0, y0):
    n = int((t2 - t1) / h + 1)
    t = np.linspace(t1, t2, n)
    x = np.zeros(n)
    y = np.zeros(n)
    x[0] = x0
    y[0] = y0
```

```
    for i in range(n - 1):
        xk1 = fx_test(t[i], x[i], y[i])
        yk1 = fy_test(t[i], x[i], y[i])
        xk2 = fx_test(t[i] + h / 3, x[i] + h * xk1/3,
        y[i] + h * yk1 / 3)
        yk2 = fy_test(t[i] + h / 3, x[i] + h * xk1/3,
        y[i] + h * yk1 / 3)
        xk3 = fx_test(t[i] + 2 * h / 3, x[i] - h * xk1/3
        + h * xk2, y[i] - h * yk1 / 3 + h * yk2)
        yk3 = fy_test(t[i] + 2 * h / 3, x[i] - h * xk1/3
        + h * xk2, y[i] - h * yk1 / 3 + h * yk2)
        xk4 = fx_test(t[i] + h, x[i] + h * (xk1 - xk2 + xk3),
        y[i] + h * (yk1 - yk2 + yk3))
        yk4 = fy_test(t[i] + h, x[i] + h * (xk1 - xk2 + xk3),
        y[i] + h * (yk1 - yk2 + yk3))
        x[i + 1] = x[i] + h * (xk1 + 3 * xk2 + 3 * xk3 + xk4)/8
        y[i + 1] = y[i] + h * (yk1 + 3 * yk2 + 3 * yk3 + yk4)/8

    return t, x, y

def solve_dynamic(t1, t2, h, x0, y0, alfa):
    n = int((t2 - t1) / h + 1)
    t = np.linspace(t1, t2, n)
    x = np.zeros(n)
    y = np.zeros(n)
    x[0] = x0
    y[0] = y0

    for i in range(n - 1):
```

```
        xk1 = fx_dynamic(t[i], x[i], y[i], alfa)
        yk1 = fy_dynamic(t[i], x[i], y[i], alfa)
        xk2 = fx_dynamic(t[i] + h / 3, x[i] + h * xk1/3,
        y[i] + h * yk1 / 3, alfa)
        yk2 = fy_dynamic(t[i] + h / 3, x[i] + h * xk1/3,
        y[i] + h * yk1 / 3, alfa)
        xk3 = fx_dynamic(t[i] + 2 * h / 3, x[i] - h * xk1/3
        + h * xk2, y[i] - h * yk1 / 3 + h * yk2, alfa)
        yk3 = fy_dynamic(t[i] + 2 * h / 3, x[i] - h * xk1/3
        + h * xk2, y[i] - h * yk1 / 3 + h * yk2, alfa)
        xk4 = fx_dynamic(t[i] + h, x[i] + h * (xk1 - xk2
        + xk3), y[i] + h * (yk1 - yk2 + yk3), alfa)
        yk4 = fy_dynamic(t[i] + h, x[i] + h * (xk1 - xk2
        + xk3), y[i] + h * (yk1 - yk2 + yk3), alfa)
        x[i + 1] = x[i] + h * (xk1 + 3 * xk2 + 3 * xk3 + xk4)/8
        y[i + 1] = y[i] + h * (yk1 + 3 * yk2 + 3 * yk3 + yk4)/8

    return t, x, y


def exact(t1, t2, h):
    n = int((t2 - t1) / h + 1)
    t = np.linspace(t1, t2, n)
    x = np.zeros(n)
    y = np.zeros(n)
    for i in range(n):
        x[i] = np.cos(t[i]**2) * np.sqrt(1 + t[i])
        y[i] = np.sin(t[i]**2) * np.sqrt(1 + t[i])
    return x, y
```

```
def main_test():
    t1 = 0
    t2 = 2
    x0 = 1
    y0 = 0
    n = 36

    # Решение методом Рунге-Кутты 4-го порядка
    N = 100
    h = (t2 - t1) / N
    t, x, y = solve_test(t1, t2, h, x0, y0)

    plt.figure()
    plt.plot(t, x, label='x(t)')
    plt.plot(t, y, label='y(t)')
    plt.legend()
    plt.savefig('1.png')
    plt.show()

    # Точное решение
    x1, y1 = exact(t1, t2, h)
    plt.figure()
    plt.plot(t, x1, label='xExact(t)')
    plt.plot(t, y1, label='yExact(t)')
    plt.legend()
    plt.savefig('2.png')
    plt.show()
```

```
# Overlay plots
plt.figure()
plt.plot(t, x, label='x(t) - RK4')
plt.plot(t, y, label='y(t) - RK4')
plt.plot(t, x1, linestyle='--', label='xExact(t)')
plt.plot(t, y1, linestyle='--', label='yExact(t)')
plt.legend()
plt.savefig('12overlay.png')
plt.show()

# Погрешность
N = np.zeros(n)
e = np.zeros(n)
h = np.zeros(n)
e4 = np.zeros(n)

for i in range(25):
    N[i] = i + 1
for i in range(25, n):
    N[i] = N[i - 1] + 25
for i in range(n):
    h[i] = (t2 - t1) / N[i]

for i in range(n):
    t, x, y = solve_test(t1, t2, h[i], x0, y0)
    x1, y1 = exact(t1, t2, h[i])
    x2 = np.abs(x - x1)
    y2 = np.abs(y - y1)
    maxx = np.max(x2)
```

```
        maxy = np.max(y2)
        e[i] = max(maxx, maxy)

    for i in range(len(e)):
        h4 = h[i]**4
        e4[i] = e[i] / h4

    # Графики
    plt.figure()
    plt.plot(N, e, label='e')
    plt.legend()
    plt.savefig('3.png')
    plt.show()

    plt.figure()
    plt.plot(N, e4, label='e/h^4')
    plt.legend()
    plt.savefig('4.png')
    plt.show()

def main_dynamic():
    t1 = 0
    t2 = 100
    x0 = 3
    y0 = 1
    alfa = 0.1

    for i in range(9):
        t, x, y = solve_dynamic(t1, t2, 0.1, x0, y0, alfa)
```

```python
        plt.figure()
        plt.plot(x, y, label='(X,Y)')
        plt.legend()
        plt.savefig(f'{i + 5}.png')
        plt.show()

        alfa += 0.1

    alfa = 0.1

    for i in range(9):
        t, x, y = solve_dynamic(t1, t2, 0.1, x0, y0, alfa)

        plt.figure()
        plt.plot(t, x, label='X(t)')
        plt.plot(t, y, label='Y(t)')
        plt.legend()
        plt.savefig(f'n{i + 5}.png')
        plt.show()

        alfa += 0.1

if __name__ == "__main__":
    main_test()
    main_dynamic()
```