Московский государственный университет
имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики

Отчет по заданию

«Численное решение модифицированной задачи хищник-жертва. Метод Рунге — Кутты»

Выполнил:
студент 305 группы
Якушев П. А.

Москва 2021

Содержание

Постановка задачи

- Задача заключается в нахождении решения системы обыкновенных дифференциальных уравнений первого порядка методом Рунге — Кутты.

$$\begin{cases} \dot{y_1} = y_1(y_1 - B)(1 - y_1) - y_1 y_3 - A(y_1 - y_2), \\ \dot{y_2} = y_2(y_2 - B)(1 - y_2) - y_2 y_3 - A(y_2 - y_1), \\ \dot{y_3} = y_3(y_1 + y_2 - C). \end{cases}$$

- Нахождение стационарной точки.
  $\dot{y_i} = 0$, $i = \overline{1..3}$
  Получим:

$$\begin{cases} 0 = y_1(y_1 - B)(1 - y_1) - y_1 y_3 - A(y_1 - y_2), \\ 0 = y_2(y_2 - B)(1 - y_2) - y_2 y_3 - A(y_2 - y_1), \\ 0 = y_3(y_1 + y_2 - C). \end{cases}$$

  Из системы получаем решение:

$$\begin{cases} y_1 = \frac{C}{2}, \\ y_2 = \frac{C}{2}, \\ y_3 = \frac{(1+B)C}{2} - B - \frac{C^2}{4}. \end{cases}$$

- Реализация численного метода для решения данной задачи.

- Изучение динамики системы с начальными условиями: $y_i(0) = y^* + \varepsilon$, где $0 < \varepsilon < 1$, $\mathrm{i} = \overline{1..3}$ с различными $C \in [0, 1]$.

Реализация численного метода для решения данной задачи

Будем использовать метод Рунге-Кутты 4-го порядка точности:
$y_{ik} = y_{i-1k} + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$, где k_1, k_2, k_3, k_4
вычисляются следующим образом:
$k_1 = f_i(t, y_1, y_2, y_3)$, $i = \overline{1..3}$
$k_2 = f_i(t + \frac{h}{2}, y_1 + k_1\frac{h}{2}, y_2 + k_1\frac{h}{2}, y_3 + k_1\frac{h}{2})$
$k_3 = f_i(t + \frac{h}{2}, y_1 + k_2\frac{h}{2}, y_2 + k_2\frac{h}{2}, y_3 + k_2\frac{h}{2})$
$k_4 = f_i(t + h, y_1 + k_3h, y_2 + k_3h, y_3 + k_3h)$
Последовательно вычисляются коэффициенты: k_1 – в исходной точке, k_2 – на половинном шаге, k_3 – тоже на половинном шаге, но по уточнённому значению k_2 вместо k_1, k_4 – на целом шаге по предыдущему значению k_3. Стоит отметить, что получаемые здесь на каждом этапе k_1, k_2, k_3, k_4 – угловые коэффициенты четырёх разных интегральных кривых в трёх точках: $t_i, t_i + \frac{h}{2}, t_{i+1} = t_i + h$. Для получения нового значения искомой функции на полном шаге используется взвешенное среднее этих коэффициентов.

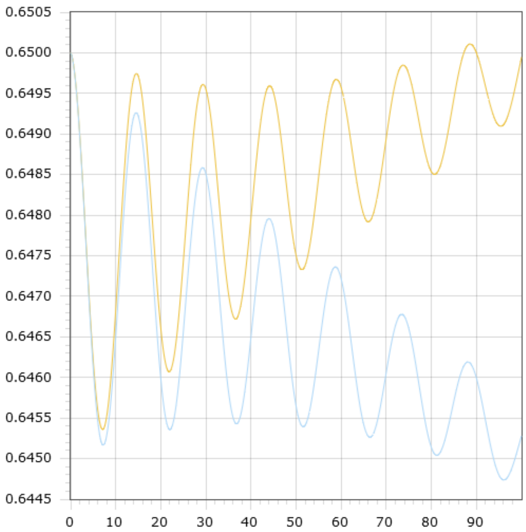
Численное решение задачи

Решим исходню систему для заданных параметров: $A = -0.02, B = 0.25$, $C \in [1.0, 2.0], \Delta C = 0.1, h = 0.01$ и начальных условиях:

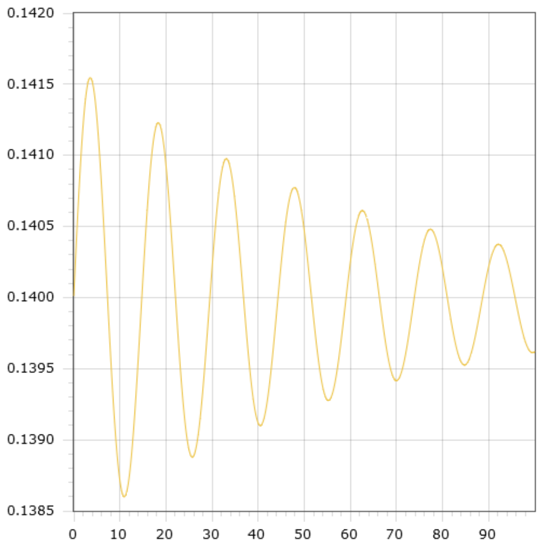
$$\begin{cases} y_1 = \frac{C}{2} + \varepsilon, \\ y_2 = \frac{C}{2} + \varepsilon, \\ y_3 = \frac{(1+B)C}{2} - B - \frac{C^2}{4} + \varepsilon. \end{cases}$$

Решим систему для $C = 1.3$

Наблюдаем квазипериодичную картину с затуханием:



$y_1(t), y_2(t)$



$y_3(t)$

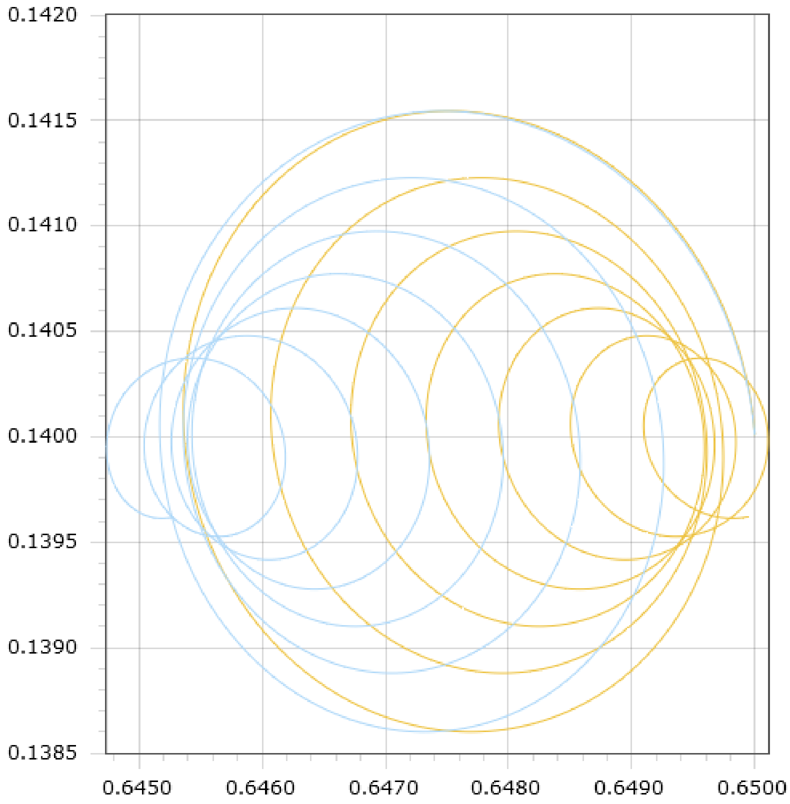
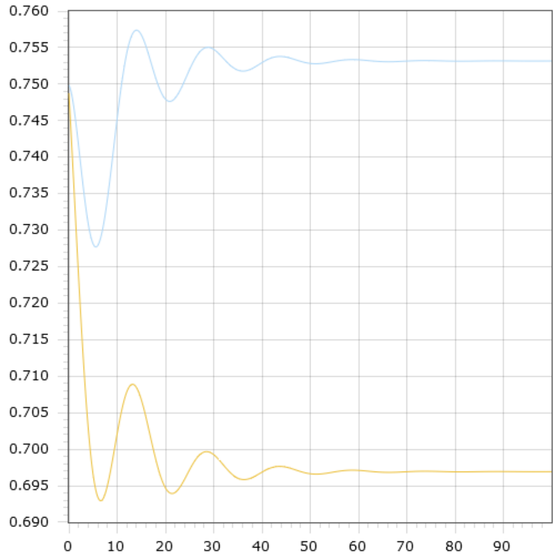


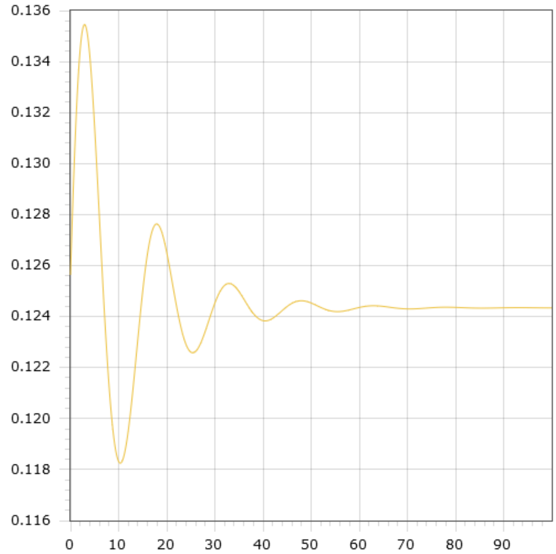
Рис. 1: $y_3(y_1), y_3(y_2)$

Решим систему для $C = 1.5$

Затухание усиливается:



$y_1(t), y_2(t)$



$y_3(t)$

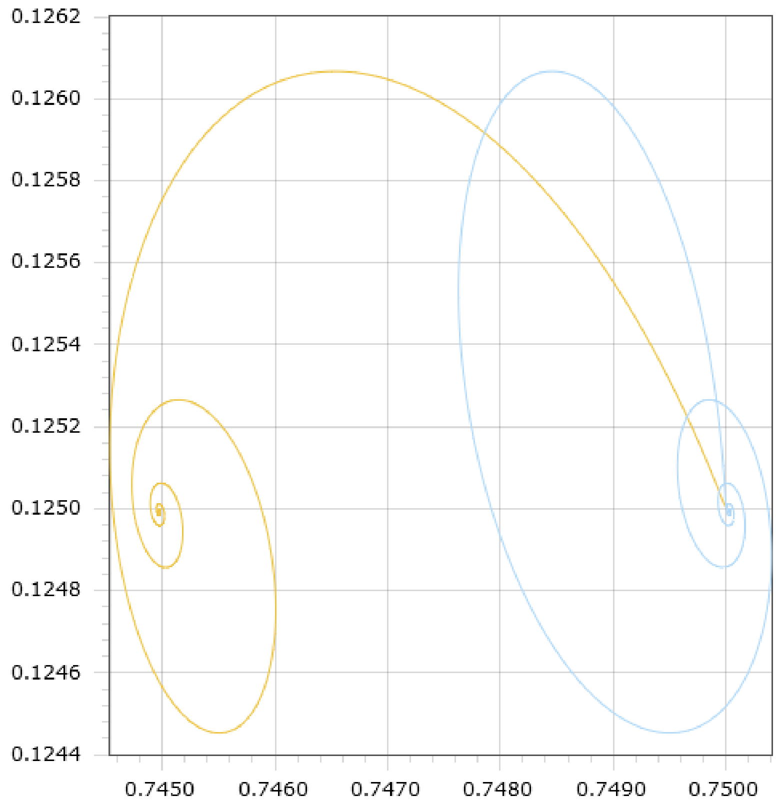
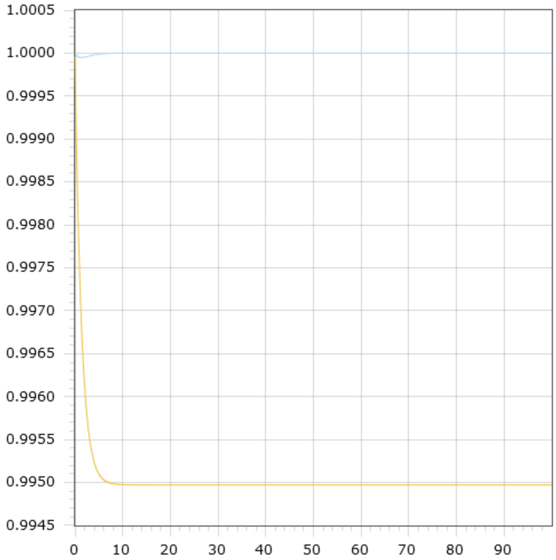


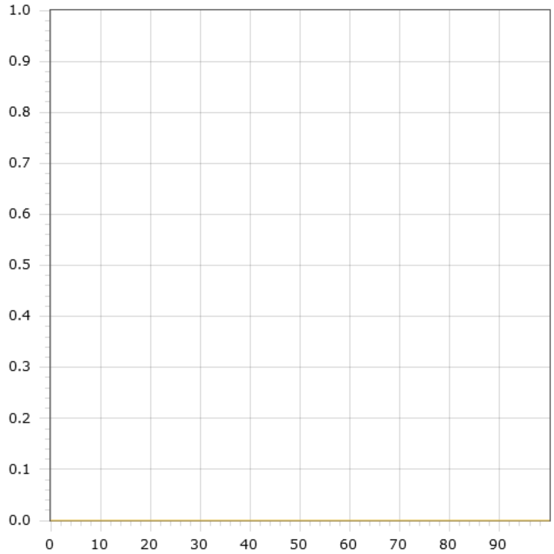
Рис. 2: $y_3(y_1), y_3(y_2)$

Решим систему для $C = 2.0$

y_3 вырождается в прямую:



$y_1(t), y_2(t)$



$y_3(t)$

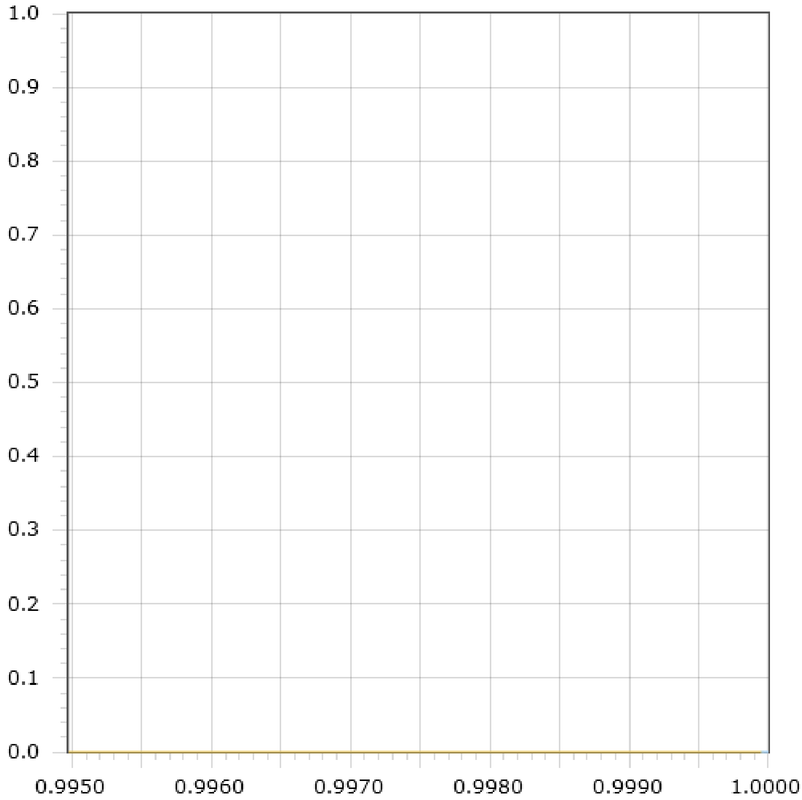


Рис. 3: $y_3(y_1), y_3(y_2)$

Исходный код

```
A = -0.02
B = 0.25
C = 1.5
h = 0.01

def f1(y1, y2, y3):
    return y1 * (y1 - B) * (1 - y1) - y1 * y3 - A * (y1 - y2)


def f2(y1, y2, y3):
    return y2 * (y2 - B) * (1 - y2) - y2 * y3 - A * (y2 - y1)


def f3(y1, y2, y3):
    return y3 * (y1 + y2 - C)


def k1(func, x, y, z):
    return func(x, y, z)


def k2(func, x, y, z, k):
    return func(x + h / 2, y + k * h / 2, z + k * h / 2)


def k3(func, x, y, z, k2):
    return func(x + h / 2, y + k2 * h / 2, z + k2 * h / 2)


def k4(func, x, y, z, k3):
    return func(x + h, y + h * k3, z + h * k3)


def next_step_value(y_prev, k1, k2, k3, k4):
    return y_prev + h * (k1 + 2 * k2 + 2 * k3 + k4) / 6


y1 = C / 2
y2 = C / 2
y3 = (1 + B) * (y1 ** 2 + y2 ** 2) / C - B - (y1 ** 2 - y1 * y2
    + y2 ** 2)
t = 0
with open('f1.csv', 'w') as p1:
    with open('f2.csv', 'w') as p2:
        with open('f3.csv', 'w') as p3:
            with open('f12.csv', 'w') as p12:
                with open('f13.csv', 'w') as p13:
                    with open('f23.csv', 'w') as p23:
                        while (t <= 100.0):
                            curr_k1 = k1(f1, y1, y2, y3)
                            curr_k2 = k2(f1, y1, y2, y3,
   curr_k1)
```

```
                curr_k3 = k3(f1, y1, y2, y3, curr_k2)
                curr_k4 = k4(f1, y1, y2, y3, curr_k3)

                y1_coeffs = [curr_k1, curr_k2, curr_k3, curr_k4]

                curr_k1 = k1(f2, y1, y2, y3)
                curr_k2 = k2(f2, y1, y2, y3, curr_k1)
                curr_k3 = k3(f2, y1, y2, y3, curr_k2)
                curr_k4 = k4(f2, y1, y2, y3, curr_k3)

                y2_coeffs = [curr_k1, curr_k2, curr_k3, curr_k4]

                curr_k1 = k1(f3, y1, y2, y3)
                curr_k2 = k2(f3, y1, y2, y3, curr_k1)
                curr_k3 = k3(f3, y1, y2, y3, curr_k2)
                curr_k4 = k4(f3, y1, y2, y3, curr_k3)

                y3_coeffs = [curr_k1, curr_k2, curr_k3, curr_k4]

                y1 = next_step_value(y1, y1_coeffs[0], y1_coeffs[1], y1_coeffs[2], y1_coeffs[3])

                y2 = next_step_value(y2, y2_coeffs[0], y2_coeffs[1], y2_coeffs[2], y2_coeffs[3])

                y3 = next_step_value(y3, y3_coeffs[0], y3_coeffs[1], y3_coeffs[2], y3_coeffs[3])

                p1.write(f"{t},{y1} \n")
                p2.write(f"{t},{y2} \n")
                p3.write(f"{t},{y3} \n")

                p12.write(f"{y1},{y2} \n")
                p13.write(f"{y1},{y3} \n")
                p23.write(f"{y2},{y3} \n")
                t += h
```
Листинг 1: Исходный код программы