

```

1  #! python3.7
2  # -*- coding: utf-8 -*-
3  from numpy import zeros, linspace
4  from matplotlib.pyplot import style, figure, axes
5
6  # Функция f подготавливает массив, содержащий элементы вектор-функции,
7  # определяющей правую часть решаемой системы ОДУ
8  def f(u, lambda) :
9      f = zeros(2)
10     f[0] = lambda*u[0]*(u[1] - u[0])
11     f[1] = 1
12     return f
13
14 # Определение входных данных задачи
15 t_0 = -1.; T = 2.
16 u_0 = 3.; lambda = 10.
17
18 # Определение числа интервалов сетки,
19 # на которой будет искомое приближённое решение
20 M = 250
21
22 # Определение сетки
23 tau = (T - t_0)/M
24 t = linspace(t_0, T, M + 1)
25
26 # Выделение памяти под массив сеточных значений решения системы ОДУ
27 # В строке с номером m этого массива хранятся сеточные значения решения,
28 # соответствующие моменту времени t_m
29 u = zeros((M + 1, 2))
30
31 # Задание начальных условий
32 # (записываются в строку с номером 0 массива u)
33 u[0] = [u_0, t_0]
34
35 # Реализация схемы Эйлера
36 # for m in range(M):
37 #     u[m + 1] = u[m] + tau*f(u[m], lambda)
38
39 # Реализация схемы ERK2
40 for m in range(M):
41     w_1 = f(u[m], lambda)
42     w_2 = f(u[m] + tau*2/3*w_1, lambda)
43     u[m + 1] = u[m] + tau*(1/4*w_1 + 3/4*w_2)
44
45 # Отрисовка решения
46 style.use('dark_background')
47
48 ax = axes(xlim=(-1, 2), ylim=(0, 3))
49 ax.set_aspect('equal'); ax.set_xlabel('t'); ax.set_ylabel('u');
50 ax.plot(u[:, 1], u[:, 0], '-y', lw=3)
51 ax.set_title('График u(t)')
52
53 # Листинг программы, реализующей решение автономной системы ОДУ
54 # после проведения процедуры автономизации

```