

```

1  from numpy import *
2  from matplotlib.pyplot import *
3
4  # Функция f подготавливает массив, содержащий элементы вектор-функции,
5  # определяющей правую часть решаемой системы ОДУ
6  def f(u,lambda):
7      f = zeros(2)
8      f[0] = lambda*u[0]*(u[1] - u[0])
9      f[1] = 1
10     return f
11
12     # Определение входных данных задачи
13     t_0 = -1.; T = 2.
14     u_0 = 3.; lambda = 10.;
15
16     # Определение числа интервалов сетки,
17     # на которой будет искаться приближённое решение
18     M = 250;
19
20     # Определение сетки
21     tau = (T - t_0)/M
22     t = linspace(t_0,T,M + 1)
23
24     # Выделение памяти под массив сеточных значений решения системы ОДУ
25     # В строке с номером m этого массива хранятся сеточные значения решения,
26     # соответствующие моменту времени t_m
27     u = zeros((M + 1,2))
28
29     # Задание начальных условий
30     # (записываются строку с номером 0 массива u)
31     u[0,0] = u_0; u[0,1] = t_0
32
33     # Реализация схемы Эйлера
34     for m in range(M):
35         u[m + 1] = u[m] + tau*f(u[m],lambda)
36
37     # Отрисовка решения
38     figure()
39     plot(u[:,1],u[:,0], '-ro', markersize=5)
40     title('График u(t)')
41     xlabel('t'); ylabel('u')
42     xlim((-1,2)); ylim((0,3))

```

Комментарий к файлу:

Листинг программы, реализующей решение системы ОДУ, возникшей после проведения процедуры автономизации задачи Коши для модельного неавтономного ОДУ. Для простоты реализована схема Эйлера. Возможно обобщение на случай использования схемы ERKs из файла Example_2.