

```

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,t,m_sun,G) :
9      f = zeros(4)
10     f[0] = u[2]
11     f[1] = u[3]
12     f[2] = - G*m_sun*u[0]/(u[0]**2 + u[1]**2)**(3/2)
13     f[3] = - G*m_sun*u[1]/(u[0]**2 + u[1]**2)**(3/2)
14     return f
15
16 # Определение входных данных задачи
17 t_0 = 0.; T = 365.25*24*60*60
18 x_0 = 147098291*10**3; y_0 = 0.
19 v_x_0 = 0.; v_y_0 = 30.4*10**3
20 G = 6.674301515151515*10**(-11)
21 m_sun = 1.98847*10**30
22
23 # Определение числа интервалов сетки,
24 # на которой будет искомое приближённое решение
25 M = 365
26
27 # Определение сетки
28 tau = (T - t_0)/M
29 t = linspace(t_0,T,M + 1)
30
31 # Выделение памяти под массив сеточных значений решения системы ОДУ
32 # В строке с номером m этого массива хранятся сеточные значения решения,
33 # соответствующие моменту времени t_m
34 u = zeros((M + 1,4))
35
36 # Задание начальных условий
37 # (записываются в строку с номером 0 массива u)
38 u[0] = [x_0, y_0, v_x_0, v_y_0]
39
40 # Реализация схемы Эйлера
41 for m in range(M) :
42     u[m + 1] = u[m] + tau*f(u[m],t[m],m_sun,G)
43
44 # Отрисовка решения
45 style.use('dark_background')
46
47 fig = figure()
48 ax = axes(xlim=(-2*10**11,2*10**11), ylim=(-2*10**11,2*10**11))
49 ax.set_aspect('equal'); ax.set_xlabel('x'); ax.set_ylabel('y');
50 ax.plot(0,0,'yo',markersize=15)
51 ax.plot(u[:,0],u[:,1],'-w',markersize=5)
52 ax.plot(u[M,0],u[M,1], color='w', marker='o', markersize=7)
53 ax.set_title('Траектория движения Земли')
54
55 # Листинг программы, реализующей решение системы ОДУ
56 # с помощью схемы Эйлера
57 # (на примере моделирования движения Земли вокруг Солнца)
58 # (результатом является траектория Земли)

```