

```

1  #! python3.7
2  # -*- coding: utf-8 -*-
3  from numpy import zeros, linspace
4  from matplotlib.pyplot import style, figure, axes
5  from celluloid import Camera
6
7  # Набор команд, за счёт которых анимация строится в отдельном окне
8  from IPython import get_ipython
9  get_ipython().run_line_magic('matplotlib', 'qt')
10
11 # Функция f подготавливает массив, содержащий элементы вектор-функции,
12 # определяющей правую часть решаемой системы ОДУ
13 def f(u,t,m_sun,G) :
14     f = zeros(4)
15     f[0] = u[2]
16     f[1] = u[3]
17     f[2] = - G*m_sun*u[0]/(u[0]**2 + u[1]**2)**(3/2)
18     f[3] = - G*m_sun*u[1]/(u[0]**2 + u[1]**2)**(3/2)
19     return f
20
21 # Определение входных данных задачи
22 t_0 = 0.; T = 365.25*24*60*60
23 x_0 = 147098291*10**3; y_0 = 0.
24 v_x_0 = 0.; v_y_0 = 30.4*10**3
25 G = 6.674301515151515*10**(-11)
26 m_sun = 1.98847*10**30
27
28 # Определение числа интервалов сетки,
29 # на которой будет искаться приближённое решение
30 M = 365
31
32 # Определение сетки
33 tau = (T - t_0)/M
34 t = linspace(t_0,T,M + 1)
35
36 # Выделение памяти под массив сеточных значений решения системы ОДУ
37 # В строке с номером m этого массива хранятся сеточные значения решения,
38 # соответствующие моменту времени t_m
39 u = zeros((M + 1,4))
40
41 # Задание начальных условий
42 # (записываются в строку с номером 0 массива u)
43 u[0] = [x_0, y_0, v_x_0, v_y_0]
44
45 # Реализация схемы Эйлера
46 for m in range(M) :
47     u[m + 1] = u[m] + tau*f(u[m],t[m],m_sun,G)
48
49 # Анимация отрисовки решения
50 style.use('dark_background')
51 fig = figure()
52 camera = Camera(fig)
53 ax = axes(xlim=(-2*10**11,2*10**11), ylim=(-2*10**11,2*10**11))
54 ax.set_aspect('equal'); ax.set_xlabel('x'); ax.set_ylabel('y')
55 ax.set_title('Траектория движения Земли')
56 for m in range(M + 1):
57     # Отрисовка в начале координат статичного Солнца
58     ax.plot(0,0,'yo',markersize=15)
59     # Отрисовка Земли в момент времени t[m]
60     ax.plot(u[m,0],u[m,1], color='w', marker='o', markersize=7)
61     # Отрисовка пути, пройденного Землёй к моменту времени t[m]
62     ax.plot(u[:m+1,0], u[:m+1,1], color='w', ls='--', lw=2)
63     camera.snap()
64 animation = camera.animate(interval=15, repeat=False, blit=True)

```

```
65
66 # Листинг программы, реализующей решение системы ОДУ
67 # с помощью схемы Эйлера
68 # (на примере моделирования движения Земли вокруг Солнца)
69 # (результатом является анимированное движение Земли)
```