

```

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,t,g,k,mass):
7      f = zeros(4)
8      f[0] = u[2]
9      f[1] = u[3]
10     f[2] = 0. - k/mass*sqrt(u[2]**2 + u[3]**2)*u[2]
11     f[3] = -g - k/mass*sqrt(u[2]**2 + u[3]**2)*u[3]
12     return f
13
14 # Определение входных данных задачи
15 t_0 = 0.; T = 7.
16 x_0 = 0.; y_0 = 0.
17 v_0 = 150.; alpha = pi/4
18 g = 9.81; koeff = 10.; mass = 500.
19
20 # Определение числа интервалов сетки,
21 # на которой будет искомое приближённое решение
22 M = 50;
23
24 # Определение схемы ERKs, которая будет использоваться для расчётов
25 # (далее необходимо раскомментировать требуемый набор коэффициентов)
26
27 # s = 1 # ERK1
28 # b = zeros(s); a = zeros((s,s)); c = zeros(s)
29 # b[0] = 1.; c[0] = 1/2
30
31 s = 2 # ERK2
32 b = zeros(s); a = zeros((s,s)); c = zeros(s)
33 b[0] = 1/4; b[1] = 3/4;
34 a[1,0] = 2/3; c[0] = 0.; c[1] = 2/3
35
36 # s =3 # ERK3
37 # b = zeros(s); a = zeros((s,s)); c = zeros(s)
38 # b[0] = 2/9; b[1] = 1/3; b[2] = 4/9
39 # a[1,0] = 1/2; a[2,0] = 0.; a[2,1] = 3/4; c[0] = 0.; c[1] = 1/2; c[2] = 3/4
40
41 # s = 4 # ERK4
42 # b = zeros(s); a = zeros((s,s)); c = zeros(s)
43 # b[0] = 1/6; b[1] = 1/3; b[2] = 1/3; b[3] = 1/6
44 # a[1,0] = 1/2; a[2,0] = 0.; a[2,1] = 1/2; a[3,0] = 0.; a[3,1] = 0.; a[3,2] = 1.
45 # c[0] = 0.; c[1] = 1/2; c[2] = 1/2; c[3] = 1.
46
47
48 # Определение сетки
49 tau = (T - t_0)/M
50 t = linspace(t_0,T,M + 1)
51
52 # Выделение памяти под массив сеточных значений решения системы ОДУ
53 # В строке с номером m этого массива хранятся сеточные значения решения,
54 # соответствующие моменту времени t_m
55 u = zeros((M + 1,4))
56
57 # Задание начальных условий
58 # (записываются строку с номером 0 массива u)
59 u[0,0] = x_0; u[0,1] = y_0
60 u[0,2] = v_0*cos(alpha); u[0,3] = v_0*sin(alpha)
61
62 # Реализация схемы ERKs
63 # (отметим, что во вспомогательных массивах adjustment_1,2 и w
64 # размерность 4 соответствует числу компонент вектор-функции f)
65 for m in range(M):
66     w = zeros((s,4))
67     for k in range(s):
68         adjustment_1 = zeros(4)
69         for l in range(k):

```

```
70         adjustment_1 = adjustment_1 + a[k,1]*w[1]
71         w[k] = f(u[m] + tau*adjustment_1,t[m] + tau*c[k],g,koeff,mass)
72     adjustment_2 = zeros(4)
73     for k in range(s):
74         adjustment_2 = adjustment_2 + b[k]*w[k]
75     u[m + 1] = u[m] + tau*adjustment_2
76
77     # Отрисовка решения
78     figure()
79     plot(u[:,0],u[:,1],'-ro',markersize=5)
80     title('Траектория движения тела')
81     xlabel('x'); ylabel('y')
82     xlim((0,1.62*80)); ylim((0,80))
```

Комментарий к файлу:

Листинг программы, реализующей схему Рунге-Кутты ERKs с s-стадиями для решения задачи Коши для системы ОДУ, возникающей при моделировании движения тела, брошенного под углом к горизонту в поле тяжести с учётом сопротивления воздуха.