

```

1  % Определение входных данных задачи
2  t_0 = 0;
3  x_0 = 3; y_0 = -4;
4  v_x_0 = 0; v_y_0 = 0;
5  g = 9.81; l = 5.0; mass = 1.0;
6  T = 2*(2*pi*sqrt(l/g)); % T равно двум периодам колебаний маятника
7
8  % Определение множителя Лагранжа
9  lambda_0 = (y_0*g - v_x_0^2 - v_y_0^2)*mass/(2*l^2);
10
11 % Определение числа интервалов сетки,
12 % на которой будет искомое приближённое решение
13 M = 500;
14
15 % Определение параметра схемы (нужный раскомментировать)
16 alpha = (1 + 1i)/2; %CROS1 (схема Розенброка с комплексным коэффициентом)
17 % alpha = 1; %DIRK1 (обратная схема Эйлера)
18
19 % Определение сетки
20 tau = (T - t_0)/M;
21 t = t_0:tau:T;
22
23 % Выделение памяти под массив сеточных значений решения системы ОДУ
24 % В строке с номером m этого массива хранятся сеточные значения решения,
25 % соответствующие моменту времени t_m
26 u = zeros(M + 1,5);
27
28 % Задание начальных условий
29 % (записываются в строку с номером 1 массива u)
30 u(1,:) = [x_0 y_0 v_x_0 v_y_0 lambda_0];
31
32 % Реализация схемы из семейства ROS1
33 % Конкретная схема определяется коэффициентом alpha
34 for m = 1:M
35     w_1 = (D - alpha*tau*f_u(u(m,:),mass))\f(u(m,:),g,mass,l);
36     u(m+1,:) = u(m,:) + tau*real(w_1)';
37 end
38
39 % Отрисовка решения
40 for m = 1:(M + 1)
41     % Отрисовка точки подвеса
42     plot(0,0,'or','MarkerSize',7); hold on;
43     plot([-2 2],[0 0],'-'); hold on;
44     % Отрисовка маятника
45     plot(u(m,1),u(m,2),'ob','MarkerSize',7,'LineWidth',7); hold on;
46     plot([0 u(m,1)],[0 u(m,2)],'-b','LineWidth',3); hold on;
47     axis([-5.5 5.5 -7.5 3.5]);
48     hold off; drawnow; pause(0.001);
49 end
50
51 function f_vec = f(u,g,mass,l)
52     % Функция подготавливает массив, содержащий элементы вектор-функции f,
53     % определяющей правую часть решаемой системы ОДУ
54     f_vec = zeros(5,1);
55     f_vec(1) = u(3);
56     f_vec(2) = u(4);
57     f_vec(3) = 2*u(5)*u(1)/mass;
58     f_vec(4) = -g + 2*u(5)*u(2)/mass;
59     f_vec(5) = u(1)^2 + u(2)^2 - l^2;
60 end
61
62 function D_matrix = D()
63     % Функция подготавливает массив, содержащий элементы матрицы D
64     D_matrix = zeros(5,5);
65     % Задаются ненулевые диагональные элементы матрицы D
66     for i = 1:4
67         D_matrix(i,i) = 1;
68     end
69 end

```

```
70
71 function f_u_matrix = f_u(u,mass)
72     % Функция подготавливает массив, содержащий элементы матрицы Якоби f_u
73     f_u_matrix = zeros(5,5);
74     % Задаются ненулевые компоненты матрицы Якоби
75     f_u_matrix(1,3) = 1;
76     f_u_matrix(2,4) = 1;
77     f_u_matrix(3,1) = 2*u(5)/mass;
78     f_u_matrix(3,5) = 2*u(1)/mass;
79     f_u_matrix(4,2) = 2*u(5)/mass;
80     f_u_matrix(4,5) = 2*u(2)/mass;
81     f_u_matrix(5,1) = 2*u(1);
82     f_u_matrix(5,2) = 2*u(2);
83 end
```

Комментарий к файлу:

Листинг программы, реализующей решение задачи Коши для дифференциально-алгебраической системы уравнений, возникающей при моделировании движения материальной точки под действием силы тяжести и жёстком условии связи.