

```

1  % Определение входных данных задачи
2  t_0 = 0; T = 2; u_0 = 1;
3
4  % Определение числа интервалов БАЗОВОЙ сетки,
5  % на которой будет искаться приближённое решение
6  M = 50;
7
8  % Число сеток, на которых ищется приближённое решение
9  S = 10;
10 % Коэффициент сгущения сеток
11 r = 2;
12
13 % Определение параметра схемы (нужный раскомментировать)
14 alpha = (1 + 1i)/2; p_theor = 2; % CROS1
15 % alpha = 1; p_theor = 1;      % DIRK1
16 % alpha = 1/2; p_theor = 2;    % KN
17 % alpha = 0; p_theor = 1;      % ERK1
18
19 % Выделение памяти под массивы сеточных значений
20 % решений ОДУ на разных сетках с номерами s = 1, ..., S
21 % Первый индекс - номер сетки s из последовательности
22 % сгущающихся сеток, на которых ищется решение
23 % Второй индекс определяет массив (для фиксированного s),
24 % в котором хранятся сеточные значения решения из узлов,
25 % совпадающих с узлами базовой сетки
26 array_of_u = zeros(S,M + 1);
27
28 % "Большой цикл", который пересчитывает решение S раз
29 % на последовательности сгущающихся сеток
30 % Массив сеточных значений решения содержит только
31 % сеточные значения из узлов, совпадающих с узлами базовой сетки
32 for s = 1:S
33     u = ODESolving(t_0,T,u_0,M,alpha,s,r);
34     array_of_u(s,:) = u;
35     s
36 end
37
38 % Отрисовка решения, полученного на сетке с номером S
39 % (отмечаются только узлы, совпадающие с узлами базовой сетки)
40 figure;
41 t = t_0:(T - t_0)/M:T; % Определение базовой сетки
42 plot(t,array_of_u(S,:), '-or', 'MarkerSize', 5, 'LineWidth', 1);
43 title('График u(t)');
44 xlabel('t'); ylabel('u');
45 axis([t_0 T -30 30]);
46
47 % Выделение памяти под массив значений эффективных
48 % порядков точности расчёта приближённого решения
49 % в каждом узле t_m, 1 <= m <= M (второй индекс массива),
50 % кроме t_0, так как в нём решение задано точно,
51 % и на разных сетках (первый индекс массива)
52 p_eff_ForEveryTime = zeros(S,M + 1);
53
54 % Вычисление эффективных порядков точности
55 for m = 2:(M + 1)
56
57     % Вычисление p^{eff}_{(1)}(t_m) и p^{eff}_{(2)}(t_m) невозможно
58     p_eff_ForEveryTime(1,m) = NaN;
59     p_eff_ForEveryTime(2,m) = NaN;
60
61     for s = 3:S
62         p_eff_ForEveryTime(s,1) = inf;
63         p_eff_ForEveryTime(s,m) = log(...
64             abs(array_of_u(s-1,m)-array_of_u(s-2,m)))/...

```

```

65         abs(array_of_u(s,m)-array_of_u(s-1,m)))/...
66         log(r);
67     end
68 end
69
70 % Отрисовка результатов расчётов для сетки с номером S
71 figure;
72 % Рисуются зависимость теоретического порядка точности p_theor
73 % от узла базовой сетки t_m
74 plot(t,t*0 + p_theor,'-g','MarkerSize',3); hold on;
75 % Рисуются зависимость эффективного порядка точности
76 % от узла базовой сетки
77 plot(t(2:M+1),p_eff_ForEveryTime(S,2:M + 1),...
78     '-sr','MarkerSize',5,'LineWidth',1);
79 axis([t_0 T -2.0 3.0]);
80 xlabel('t'); ylabel('p^{eff}');
81
82 function f_res = f(u)
83     % Функция f возвращает значение правой части решаемого ОДУ
84     f_res = u^2;
85 end
86
87 function f_u_res = f_u(u)
88     % Функция f_u возвращает значение частной производной f_u
89     f_u_res = 2*u;
90 end
91
92 function u_basic = ODESolving(t_0,T,u_0,M_0,alpha,s,r)
93     % Функция реализует решение ОДУ на сетке с M интервалами
94     % по схеме, определяемой параметром alpha
95
96     % Входные параметры:
97     % t_0, T - начальный и конечный моменты счёта
98     % u_0 - начальное условие
99     % M_0 - число интервалов базовой сетки по времени
100    % alpha - параметр определяющий схему (ERK1,KN,DIRK1,CROS1)
101    % s - номер сетки, на которой вычисляется решение
102    % (если s = 1, то решение ищется на базовой сетке)
103    % r - коэффициент сгущения сетки
104
105    % Выходной параметр:
106    % u_basic - массив, содержащий сеточные значения
107    % решения ОДУ только в узлах, совпадающих с узлами базовой сетки
108
109    % Формирование сгущённой в r^(s - 1) раз сетки с номером s:
110
111    % Вычисление числа интервалов на сетке с номером s
112    M = M_0*r^(s - 1);
113    % Определение шага сгущённой сетки
114    tau = (T - t_0)/M;
115    % Определение сгущённой сетки
116    t = t_0:tau:T;
117
118    % Выделение памяти под массив сеточных значений решения ОДУ,
119    % в котором будут храниться сеточные значения из узлов,
120    % совпадающих с узлами базовой сетки
121    u_basic = zeros(1,M_0 + 1);
122
123    % Выделение памяти под массив сеточных значений
124    % решения на сгущённой сетке
125    u = zeros(1,M + 1);
126
127    % Задание начального условия
128    u(1) = u_0;

```

```
129
130 % Реализация схемы из семейства ROS1
131 % Конкретная схема определяется коэффициентом alpha
132 for m = 1:M
133     w_1 = f(u(m))/(1 - alpha*tau*f_u(u(m)));
134     u(m + 1) = u(m) + tau*real(w_1);
135 end
136
137 % Из массива u выбираются сеточные значения из узлов,
138 % совпадающих с узлами базовой сетки
139 for m = 1:(M_0 + 1)
140     u_basic(1,m) = u((m - 1)*r^(s - 1) + 1);
141 end
142
143 end
```

Комментарий к файлу:

Листинг программы, реализующей вычисление эффективных порядков точности для полученного численного решения с целью определения области существования решения задачи Коши.