```
% Определение входных данных задачи
t_0 = -1; T = 2;
u_0 = 3; lambd = 10;

% Определение числа интервалов сетки,
% на которой будет искаться приближённое решение
M = 50;

% Отрисовка решения
figure()
[t,u] = ODESolving(t_0,T,u_0,lambd,M,0); % ERK1
plot(t,u,'-ro','MarkerSize',5); hold on;
[t,u] = ODESolving(t_0,T,u_0,lambd,M,0.5); % схема с полусуммой
plot(t,u,'-mo','MarkerSize',5); hold on;
[t,u] = ODESolving(t_0,T,u_0,lambd,M,1); % обратный Эйлер
plot(t,u,'-bo','MarkerSize',5); hold on;
[t,u] = ODESolving(t_0,T,u_0,lambd,M,(1 + 1i)/2); % CROS1
plot(t,u,'-go','MarkerSize',5); hold on;
title('График u(t)');
xlabel('t'); ylabel('u');
axis([t_0 T 0 u_0]);
legend('ERK1','KN','DIRK1','CROS1')

function f_res = f(u,t,lambd)
    % Функция f возвращает значение правой части решаемого ОДУ
    f_res = lambd*u*(t - u);
end

function f_u_res = f_u(u,t,lambd)
    % Функция f_u возвращает значение частной производной f_u
    f_u_res = lambd*(t - 2*u);
end

function [t,u] = ODESolving(t_0,T,u_0,lambd,M,alpha)
    % Функция реализует решение ОДУ на сетке с M интервалами
    % по схеме, определяемой параметром alpha

    % Определение сетки
    tau = (T - t_0)/M;
    t = t_0:tau:T;
    % Выделение памяти под массив сеточных значений решения ОДУ
    u = zeros(1,M + 1);
    % Задание начального условия
    u(1) = u_0;
    % Реализация схемы из семейства ROS1
    % Конкретная схема определяется коэффициентом alpha
    for m = 1:M
        w_1 = f(u(m),t(m) + tau/2,lambd)...
            /(1 - alpha*tau*f_u(u(m),t(m),lambd));
        u(m + 1) = u(m) + tau*real(w_1);
    end
end

```

Комментарий к файлу:

Листинг программы, реализующей решение модельного жёсткого ОДУ с помощью различных схем из одностадийного семейства схем Розенброка-Ванера.