```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This code is applicable to problem 2 of Homework 6
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all

% Main routine (driver) for the Oregonator
% This was built off of all the previously used code, so all functions
% worked on in this class so far can be accessed from here.
% Specify name of user supplied function M-file with rhs of ode
odefun = 'Oregonator';
% Specify the method to be used
% Options are ExplicitEuler, RK1, RK4, HW2, ODE45v4, and the default
 Matlab
% integrators
method = 'ode15s';
% Specify if you want automatic time steps, and if so, the tolerance
% autostep is only applicable as long as ODE45v4 isn't selected
autostep = true;
TOL = 1e-10;
global count;
count = 0;
% If automatic time steps are not to be used, specify the number of
 steps
NSTEP=5*(10^5);
% If ODE45v4 is to be used, state whether or not you want every step
 output
trace = 0;
% Specify the initial conditions
% Specify initial and final times
t0 = 0; tfinal = 1;
TSPAN = [t0,tfinal];
% Specify column vector of initial values
U0 = [0.06; 3.3e-7; 5.01e-11; 0.03; 2.4e-8];

% Specify any options that you want for ode45, ode23s, ode15s, etc.
% I didn't like the notion that ode45 couldn't handle the Oregonator,
 so I
% tweaked the tolerances until it worked.
OPTIONS = odeset('RelTol', 1e-8, 'AbsTol', 1e-10, 'NonNegative', 1:5);

% Build the Butcher array based on the method selected
% Only applicable as long as ODE45v4 isn't selected
% This is all probably bad programming practice, but it works and was
 easy.
if (strcmp(method, 'ExplicitEuler')|strcmp(method, 'RK1')|...
        strcmp(method, 'RK4')|strcmp(method, 'HW2') )
    if strcmp(method, 'RK4')
        A = [0 0 0 0; 0.5 0 0 0; 0 0.5 0 0; 0 0 1 0];
        b = [1/6;1/3;1/3;1/6];
        c = [0;0.5;0.5;1];
```

```
    elseif (strcmp(method,'ExplicitEuler') || strcmp(method,'RK1'))
        A = 0;
        b = 1;
        c = 0;
    elseif strcmp(method,'HW2')
        A = [0 0; 1 0];
        b = [0.5;0.5];
        c = [0;1];
    end
    % Series of if statements that will return the order of the RK
 method used
    if round(sum(b),4) == 1.0000
        qorder = 1;
        if round(b.'*c,4) == 0.5000
            qorder = 2;
            if round(b.'*(c.^2),4) == round(1/3,4) && ...
               round(sum(b'.*sum(A'.*c)),4) == round(1/6,4)
                qorder = 3;
                if round(b.'*(c.^3),4) == 1/4 && ...
                   round(sum(b'.*c'.*sum(A'.*c)),4) == 1/8 && ...
                   round(sum(b'.*sum(A'.*c.^2)),4) == round(1/12,4)
 && ...
                   round(sum(A.*b)*sum(A'.*c)',4) == round(1/24,4)
                    qorder = 4;
                end
            end
        end
    end
    % Call the solver, based on if you want automatic time step or not
    if autostep == false
        [t,U] = eulerw17d(odefun,TSPAN,U0,NSTEP,method,A,b,c);
    else
        [t,U] = RKw17sc(odefun,TSPAN,U0,TOL,A,b,c,qorder);
    end
elseif (strcmp(method, 'ODE45v4'))
    [t,U] = ode45v4(odefun,t0,tfinal,U0,TOL,trace);
elseif (strcmp(method, 'ode45'))
    [t,U] = ode45(odefun,TSPAN,U0,OPTIONS);
elseif (strcmp(method, 'ode15s'))
    [t,U] = ode15s(odefun,TSPAN,U0,OPTIONS);
elseif (strcmp(method, 'ode23s'))
    [t,U] = ode23s(odefun,TSPAN,U0,OPTIONS);
end

timesteps = length(t);

tvector = [t0:tfinal:timesteps-1];

% plot numerical solution;
figure;
hold off;
% U in the methods used earlier are annoyingly transposed...
for i = 1:5
    subplot(2,3,i)
```

```
    semilogy(t,U(:,i))
    titlestr = sprintf('c_{%i}',i);
    title(titlestr)
end
subplot(2,3,6)
plot(tvector,t)
title('t_n Vector')
titlestr = sprintf(['Method = ' method ...
    ', Time Steps = %i, Evaluations = %i'],timesteps,count);
suptitle(titlestr)
```

Published with MATLAB® R2016b