Maddox Gonzalez

Contents

Project 8

Due 4/6/25

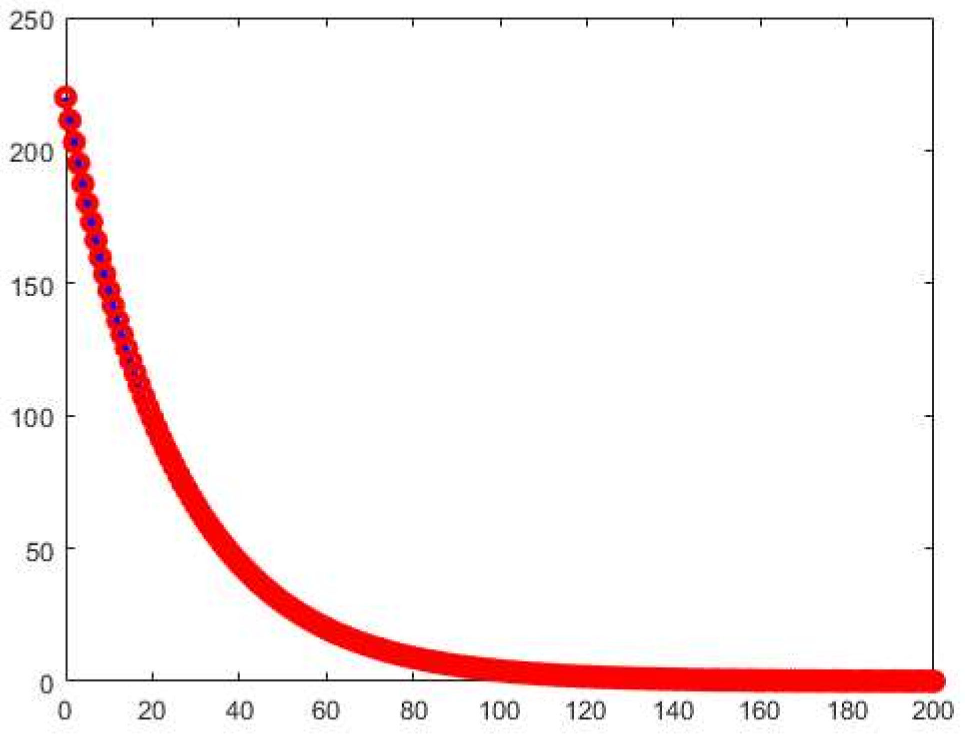
```
clc, clear, close all
% Example function: RC Circuit
% ODE is in terms of 2 variables, Voltage and time
% dV/dt + V/RC = 0
% Therefore dV/dt = -V/RC

R = 100;
C = 0.25;
dVdt = @(t,V) -V/(R*C);
Vo = 220;
tf = 200;
% exact solution
V = @(t) Vo*exp(-t/(R*C));
% solver settings
dt = [1, 0.5, 0.25];
tol = [1e-3, 1e-6];
print_time = [1, 5, 10];

for j = 1:length(dt)
    % Choosing timestep for convergence:
    % dt small enough that V doesn't overshoot the final value (0) by 2x
    % Same way that omega is always < 2
    % |V| >= |dt*V'| = dt*V/RC
    % dt <= RC
    % For strict undershoot, dt <= RC/2

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % using Euler's Method
    fprintf('Using dt = %.4f\n', dt(j));

    [time, sol] = euler_solver(dVdt, dt(j), tf, Vo);

    % Only print Euler solution for requested time points
    for i = 1:length(print_time)
        interp_sol_euler = interp1(time, sol, print_time(i));
        fprintf('Euler: %.4f at t = %.4f\n', interp_sol_euler, print_time(i));
    end

    % plot the solutions for Euler method
    figure
    plot(time, sol, 'b', 'linewidth', 3)
    hold on
    plot(time, V(time), 'ro', 'linewidth', 3)
    title(sprintf("Voltage vs. Time for dt = %.3f\n", dt(j)));
    fprintf('\n');
```

```
end

for j = 1:length(tol)
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % using adaptive Runge-Kutta 4th order method (ODE45)
    fprintf('Using tol = %.4e\n', tol(j));

    % Solve with ODE45
    [time_ode45, sol_ode45] = ode45(dVdt, 0:0.01:tf, Vo);

    % Only print ODE45 solution for requested time points
    for i = 1:length(print_time)
        interp_sol_ode45 = interp1(time_ode45, sol_ode45, print_time(i));
        fprintf('ODE45: %.4f at t = %.4f\n', interp_sol_ode45, print_time(i));
    end

    % plot the solutions for ODE45 method
    figure
    plot(time_ode45, sol_ode45, 'b', 'linewidth', 3)
    hold on
    plot(time_ode45, V(time_ode45), 'ro', 'linewidth', 3)
    title(sprintf("Voltage vs. Time for tol = %.3e\n", tol(j)));
    fprintf('\n');
end

% ==========================================================
% Function: Euler's Method Solver
% Inputs: function handle f(t, y), step size dt, final time tf, initial value vo
% Outputs: time vector t, solution vector y
function [t, y] = euler_solver(f, dt, tf, vo)
    N = ceil(tf / dt);
    t = 0:dt:tf;
    y = zeros(1, N + 1);
    y(1) = vo;
    for i = 1:N
        y(i + 1) = y(i) + dt * f(t(i), y(i));
    end
end
```

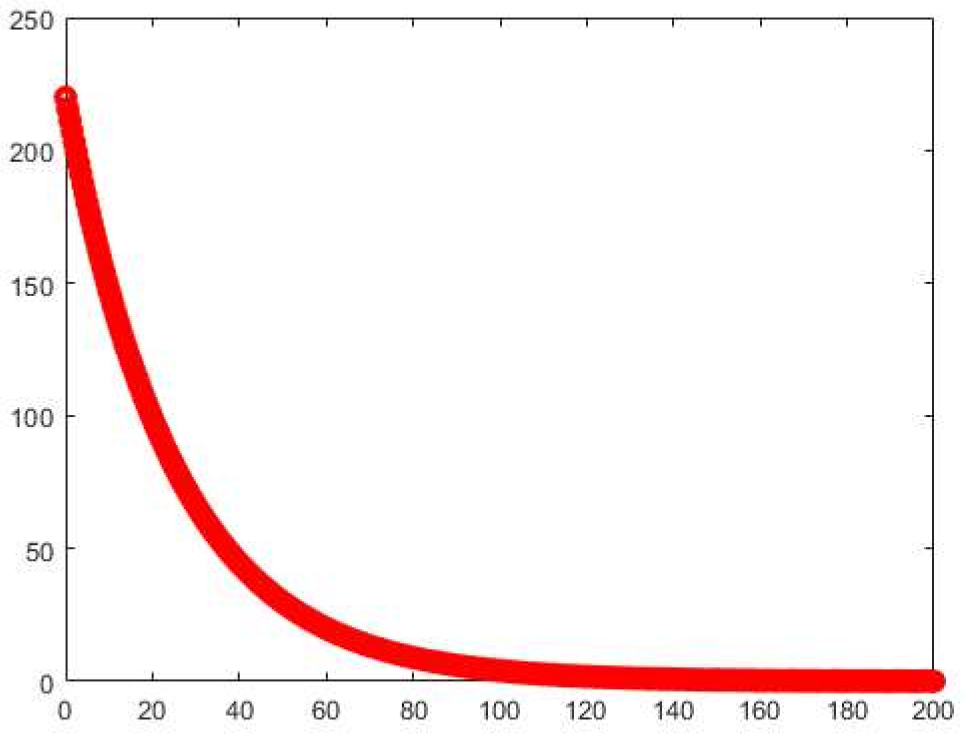
```
Using dt = 1.0000
Euler: 211.2000 at t = 1.0000
Euler: 179.3820 at t = 5.0000
Euler: 146.2632 at t = 10.0000

Using dt = 0.5000
Euler: 211.2880 at t = 1.0000
Euler: 179.7560 at t = 5.0000
Euler: 146.8738 at t = 10.0000

Using dt = 0.2500
Euler: 211.3311 at t = 1.0000
Euler: 179.9395 at t = 5.0000
Euler: 147.1738 at t = 10.0000

Using tol = 1.0000e-03
ODE45: 211.3737 at t = 1.0000
ODE45: 180.1206 at t = 5.0000
```

```
ODE45: 147.4608 at t = 10.0000

Using tol = 1.0000e-06
ODE45: 211.3737 at t = 1.0000
ODE45: 180.1206 at t = 5.0000
ODE45: 147.4608 at t = 10.0000
```

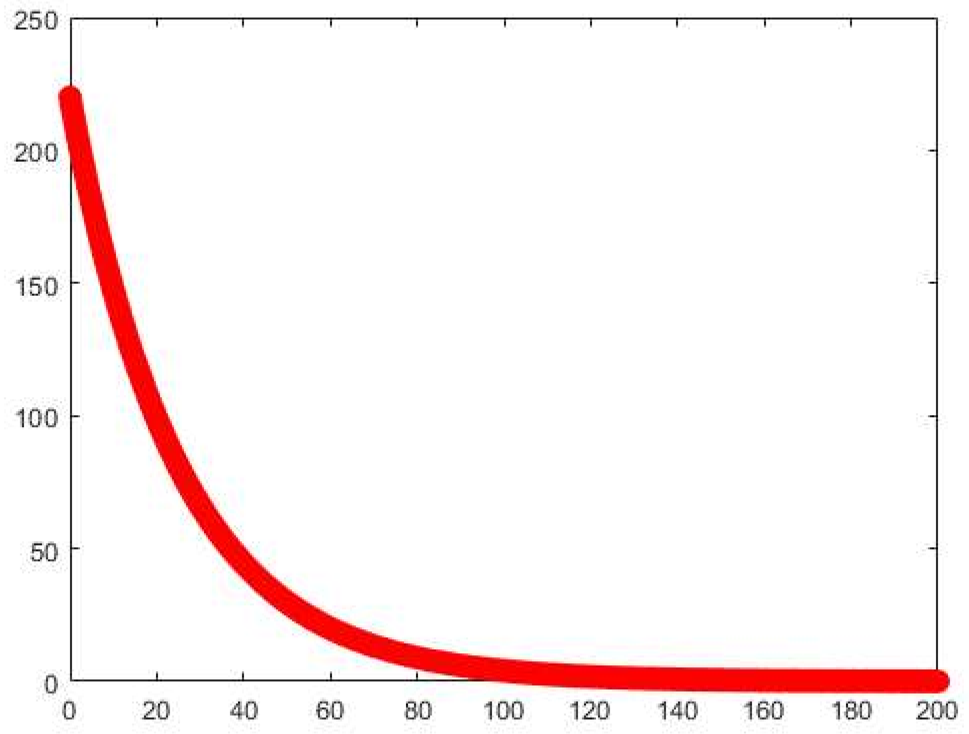
Voltage vs. Time for dt = 1.000



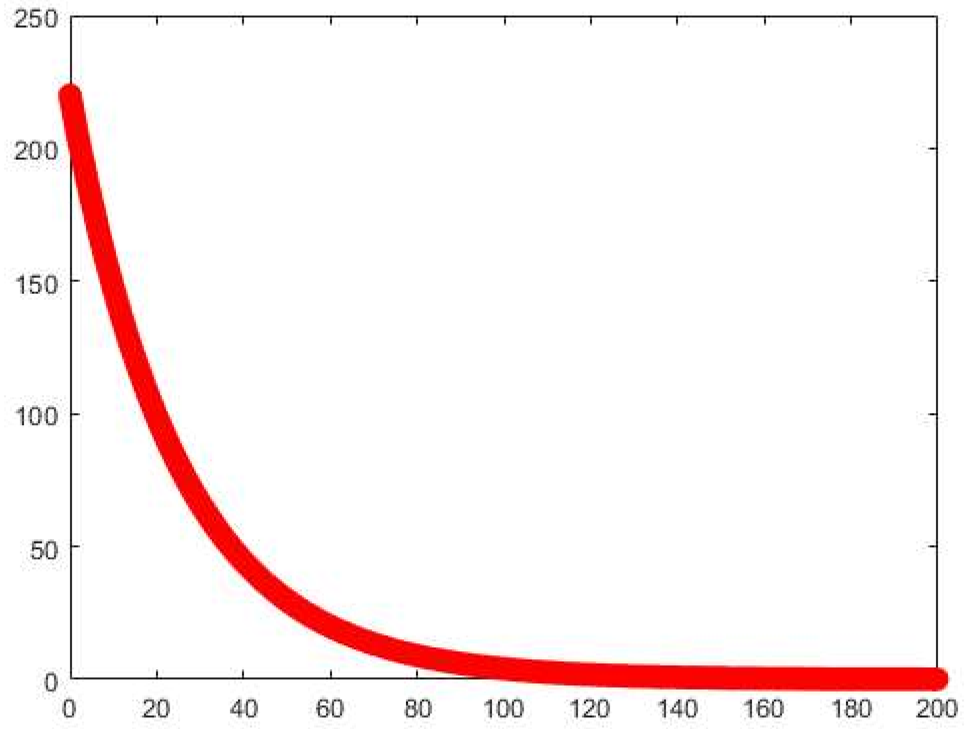
Voltage vs. Time for dt = 0.500

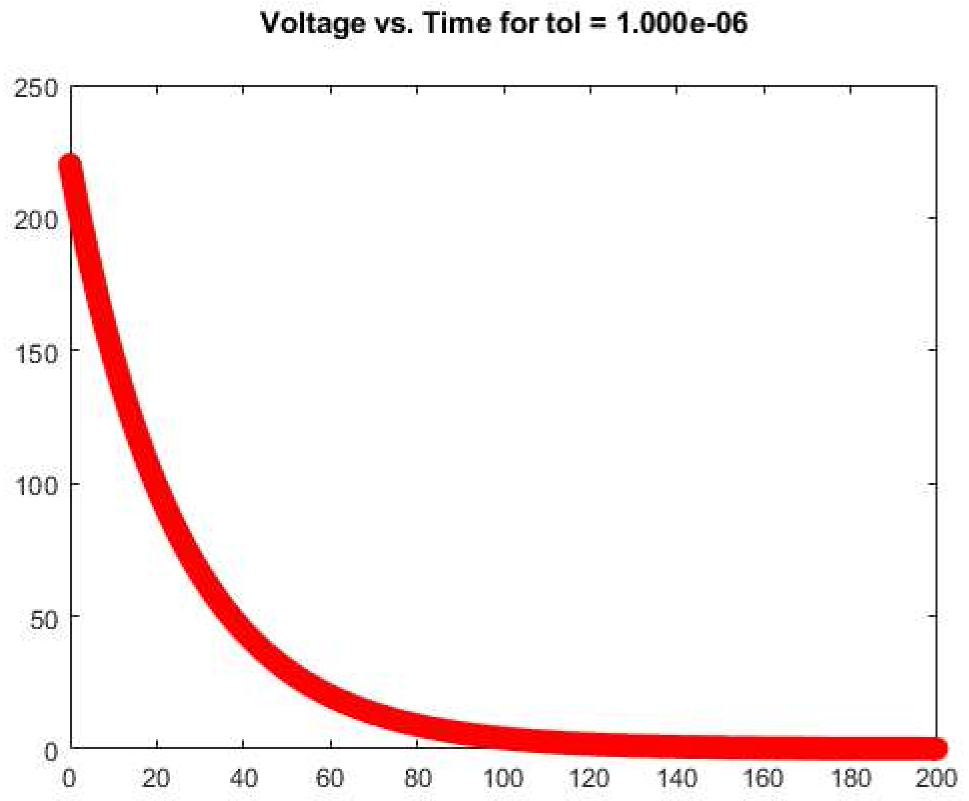


Voltage vs. Time for dt = 0.250



Voltage vs. Time for tol = 1.000e-03




Voltage vs. Time for tol = 1.000e-06
250
200
150
100
50
0
0
20
40
60
80
100
120
140
160
180
200