

Second-Order Lab: Second-Order Linear DEs in MATLAB

In this lab, you will learn how to use `iode` to plot solutions of second-order ODEs. You will also learn to classify the behaviour of different types of solutions.

Moreover, you will write your own Second-Order ODE system solver, and compare its results to those of `iode`.

Opening the m-file `lab5.m` in the MATLAB editor, step through each part using cell mode to see the results. Compare the output with the PDF, which was generated from this m-file.

There are seven (7) exercises in this lab that are to be handed in on the due date of the lab. Write your solutions in the template, including appropriate descriptions in each step. Save the m-files and submit them on Quercus.

MAT292, Fall 2019, Stinchcombe & Parsch, modified from MAT292, Fall 2018, Stinchcombe & Khovanskii, modified from MAT292, Fall 2017, Stinchcombe & Sinnamon, modified from MAT292, Fall 2015, Sousa, based on Spring 2010, Jerrard

Contents

- [Student Information](#)
- [iode for Second-Order Linear DEs with constant coefficients](#)
- [Growth and Decay Concepts](#)
- [Example](#)
- [Exercise 1](#)
- [Exercise 2](#)
- [Exercise 3](#)
- [Example](#)
- [Exercise 4](#)
- [Exercise 5](#)
- [Numerical Methods for Second-Order ODEs](#)
- [Exercise 6](#)
- [Exercise 7](#)

Student Information

Student Name: Michael Boyadjian

Student Number: 10051009142

iode for Second-Order Linear DEs with constant coefficients

In the `iode` menu, select the `Second order linear ODEs` module. It opens with a default DE and a default forcing function $f(t) = \cos(2t)$. The forcing function can be plotted along with the solution by choosing `Show forcing function` from the `Options` menu.

Use this module to easily plot solutions to these kind of equations.

There are three methods to input the initial conditions:

Method 1. Enter the values for t_0 , $x(t_0)$, and $x'(t_0)$ into the `Initial conditions` boxes, and then click `Plot solution`.

Method 2. Enter the desired slope $x'(t_0)$ into the appropriate into the **Initial conditions** box, and then click on the graph at the point $(t_0, x(t_0))$ where you want the solution to start.

Method 3. Press down the left mouse button at the desired point $(t_0, x(t_0))$ and drag the mouse a short distance at the desired slope $x'(t_0)$. When you release the mouse button, **ode** will plot the solution.

Growth and Decay Concepts

We want to classify different kinds of behaviour of the solutions. We say that a solution:

grows if its magnitude tends to infinity for large values of t , that is, if either the solution tends to $+\infty$ or $-\infty$,

decays if its magnitude converges to 0 for large values of t ,

decays while oscillating if it keeps changing sign for large values of t and the amplitude of the oscillation tends to zero,

grows while oscillating if it keeps changing sign for large values of t and the amplitude of the oscillation tends to infinity.

Example

```
t = 0:0.1:10;

% Example 1
figure();
y1 = exp(t);
plot(t,y1)

% Annotate the figure
xlabel('t');
ylabel('f_1(t)');
title('The function e^t grows');
legend('f_1(t)=e^t');

% Example 2
figure();
y2 = -exp(t);
plot(t,y2)

% Annotate the figure
xlabel('t');
ylabel('f_2(t)');
title('The function -e^{-t} decays');
legend('f_2(t)=-e^{-t}');

% Example 3
figure();
y3 = exp(-t);
plot(t,y3)

% Annotate the figure
xlabel('t');
ylabel('f_3(t)');
title('The function e^{-t} decays');
legend('f_3(t)=e^{-t}');
```

% Example 4

```
figure();  
y4 = exp(-t).*cos(t);  
plot(t,y4)
```

% Annotate the figure

```
xlabel('t');  
ylabel('f_4(t)');  
title('The function  $e^{-t}\cos(t)$  decays while oscillating');  
legend('f_4(t)= $e^{-t}\cos(t)$ ');
```

% Example 5

```
figure();  
y5 = exp(t).*sin(2*t);  
plot(t,y5)
```

% Annotate the figure

```
xlabel('t');  
ylabel('f_5(t)');  
title('The function  $e^t\sin(2t)$  grows while oscillating');  
legend('f_5(t)= $e^t\sin(2t)$ ');
```

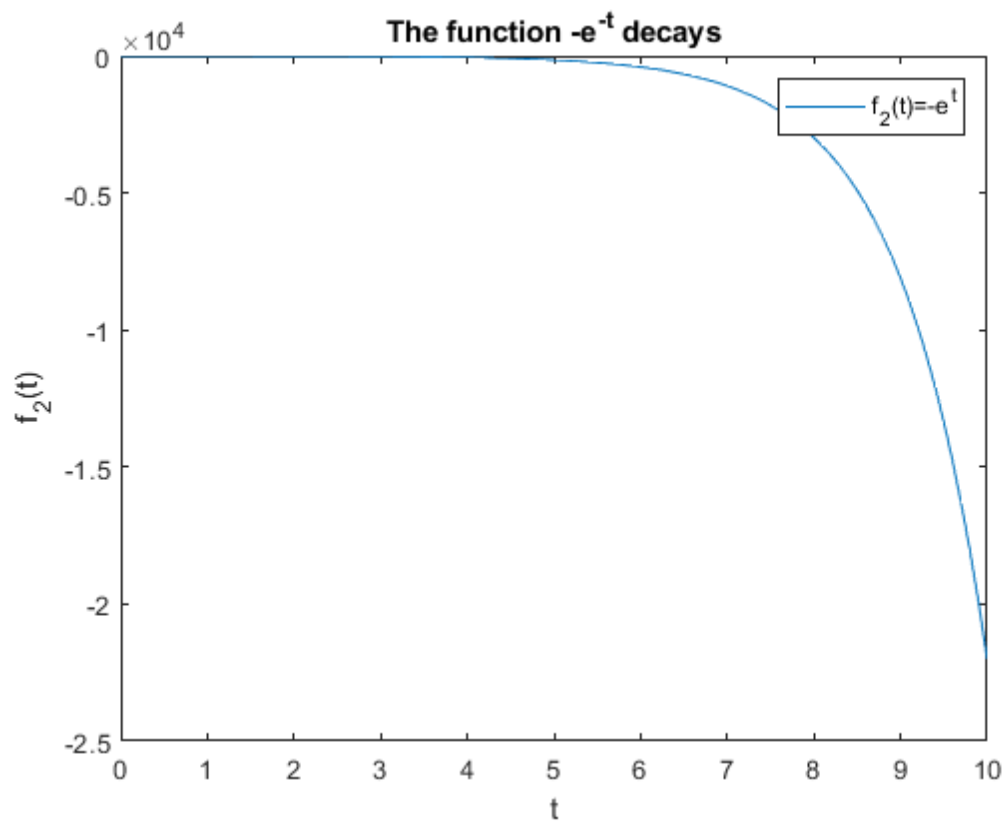
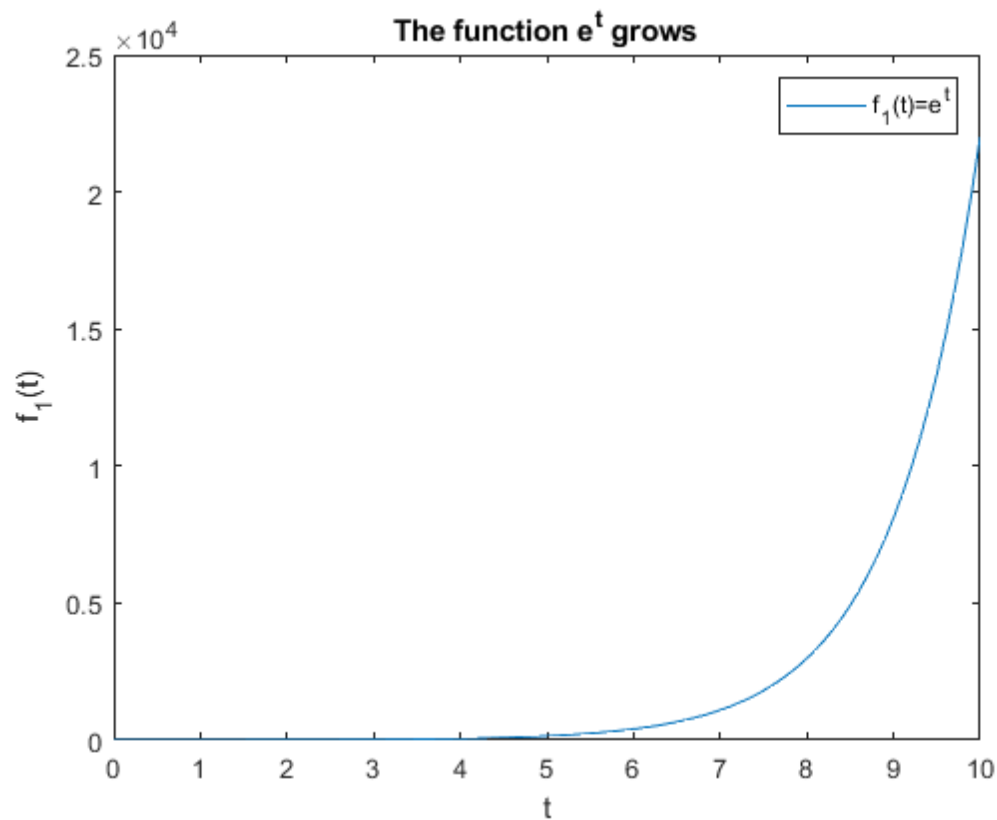
% Example 6

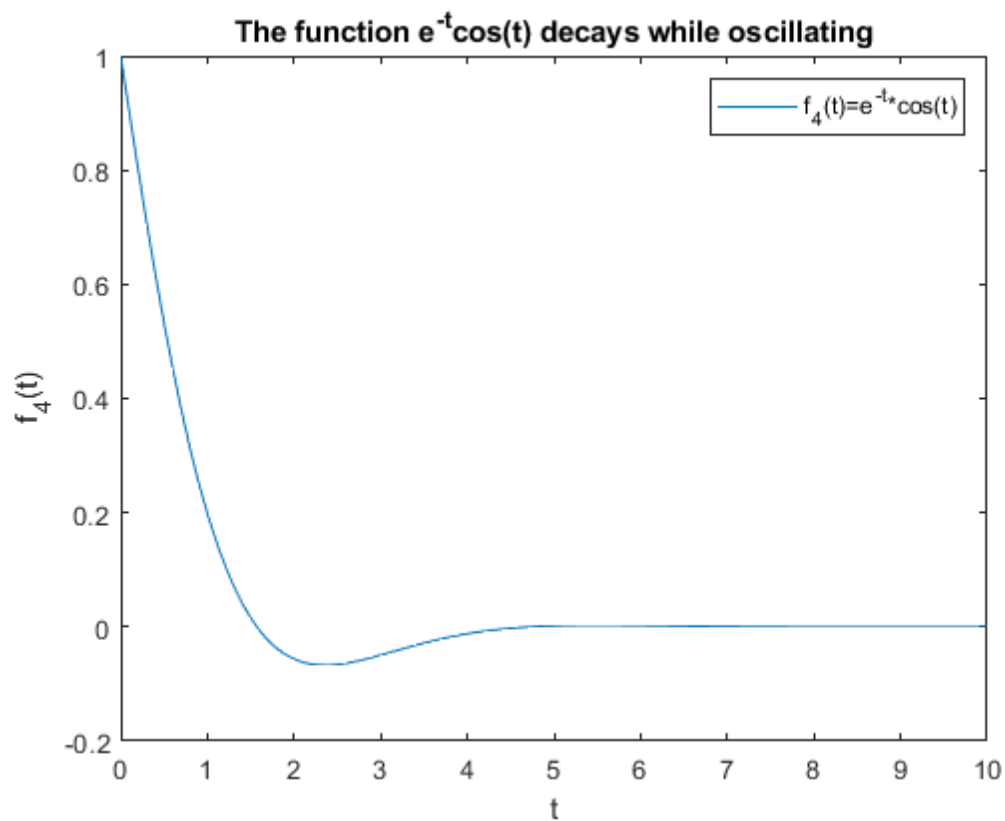
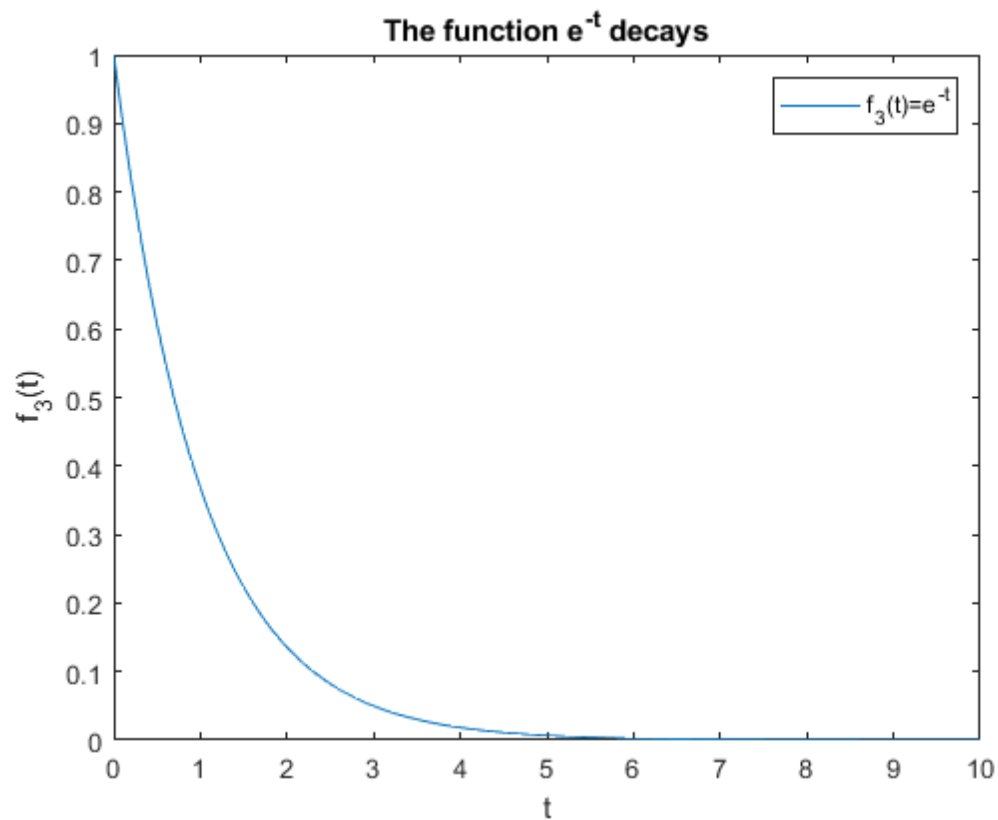
```
figure();  
y6 = sin(3*t);  
plot(t,y6)
```

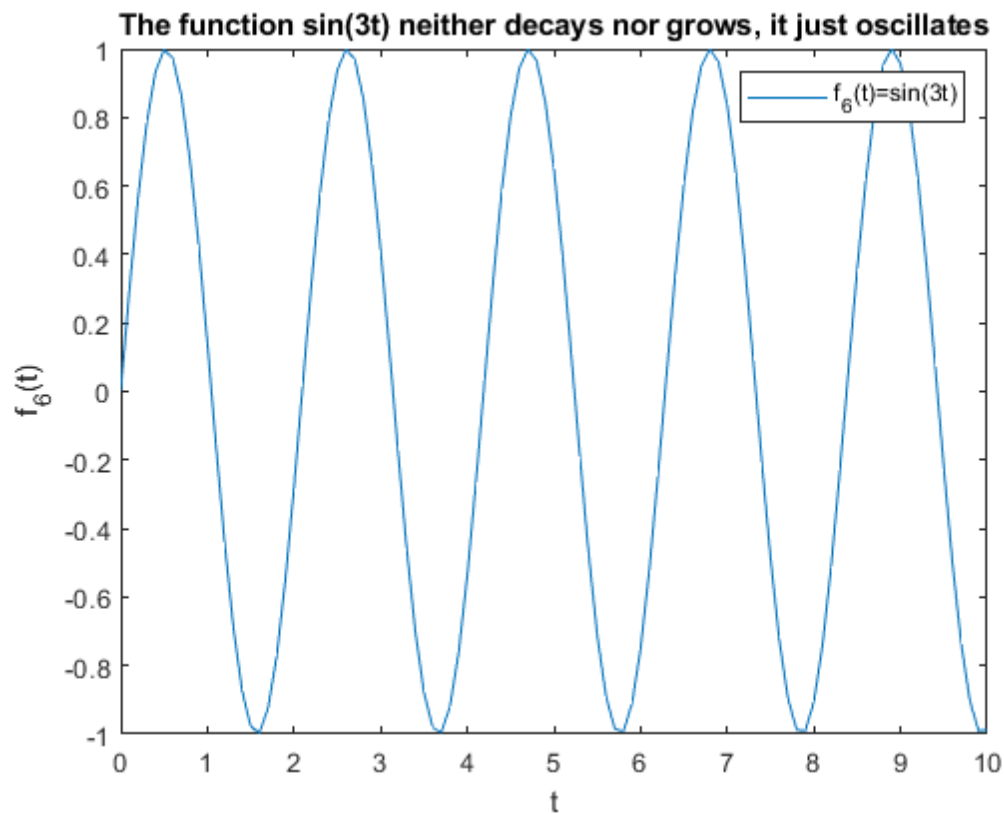
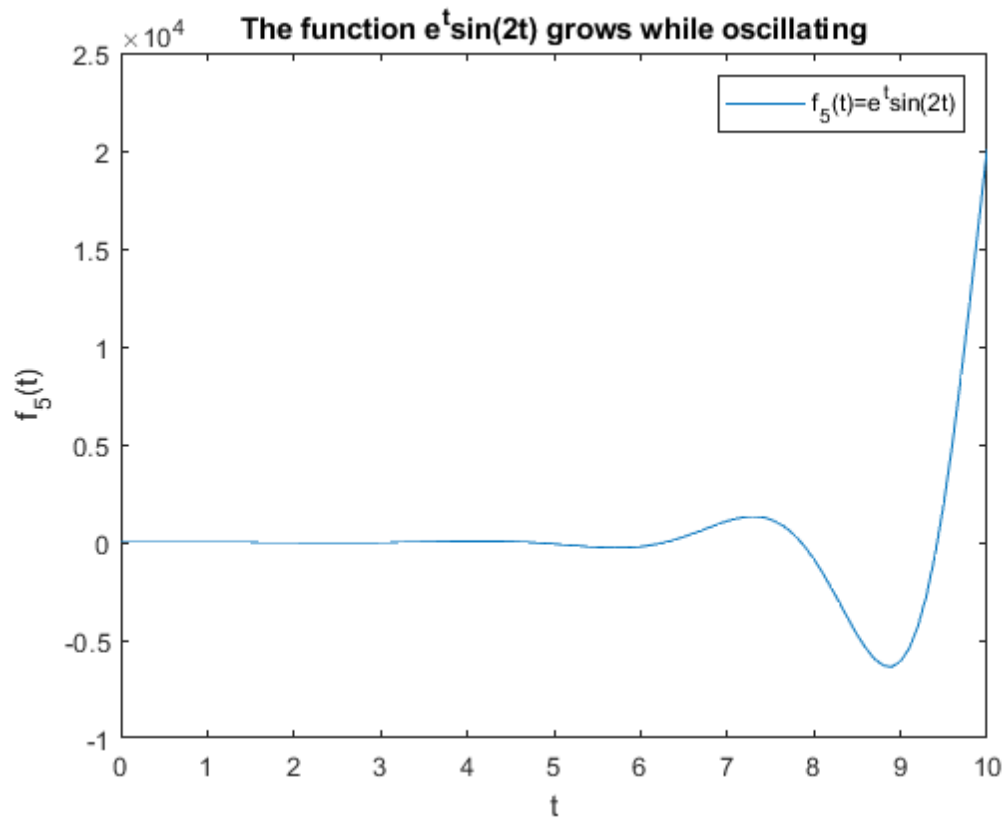
% Annotate the figure

```
xlabel('t');  
ylabel('f_6(t)');  
title('The function  $\sin(3t)$  neither decays nor grows, it just oscillates');  
legend('f_6(t)= $\sin(3t)$ ');
```

```
% |Remark.| A function which |grows while oscillating| doesn't |grow|,  
% because it keeps changing sign, so it neither tends to  $+\infty$  nor to  
%  $-\infty$ .
```







Exercise 1

Objective: Use `ode` to solve second-order linear DEs. And classify them.

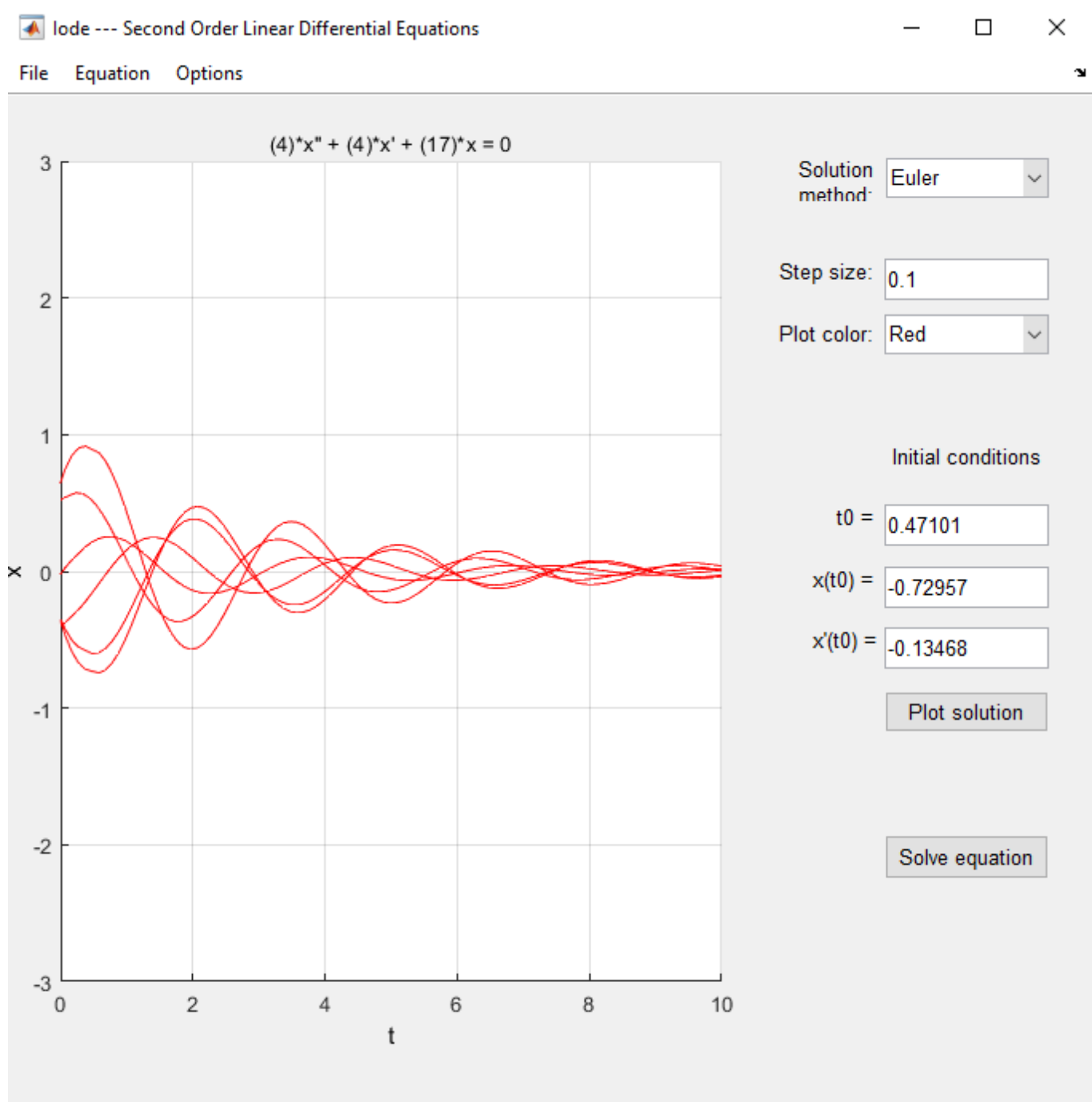
Details: Consider the ODE:

$$4y'' + 4y' + 17y = 0$$

(a) Use `iode` to plot six (6) numerical solutions of this equation with "random" initial data (use Method 3 above) and press-and-drag at various initial points, with some of the slopes being positive and some negative)

Use only initial points in the part of the window where $0 < t < 1$ and $-1 < x < 1$ and take all initial slopes between -3 and $+3$.

Change the window to $[0, 10] \times [-3, 3]$. Save a cropped screenshot with the filename `ex1_<UTORid>.png` Changing "UTORid" below will result in the image being included when you "Publish".



(b) Based on the results of (a), state what percentage of solutions decay, grow, grow while oscillating, or decay while oscillating.

As seen in the graphic, all of the solutions (100%) decay while oscillating.

(c) Solve the DE and write the exact solution. Explain why this justifies your answer in (b).

The exact solution is $y = (e^{(-t/2)}) * (c_1 \cos(2t) + c_2 \sin(2t))$. We can see that the solution decays while oscillating as the $e^{(-t/2)}$ term tends to 0 as t approaches infinity and $c_1 \cos(2t) + c_2 \sin(2t)$ oscillates for all t

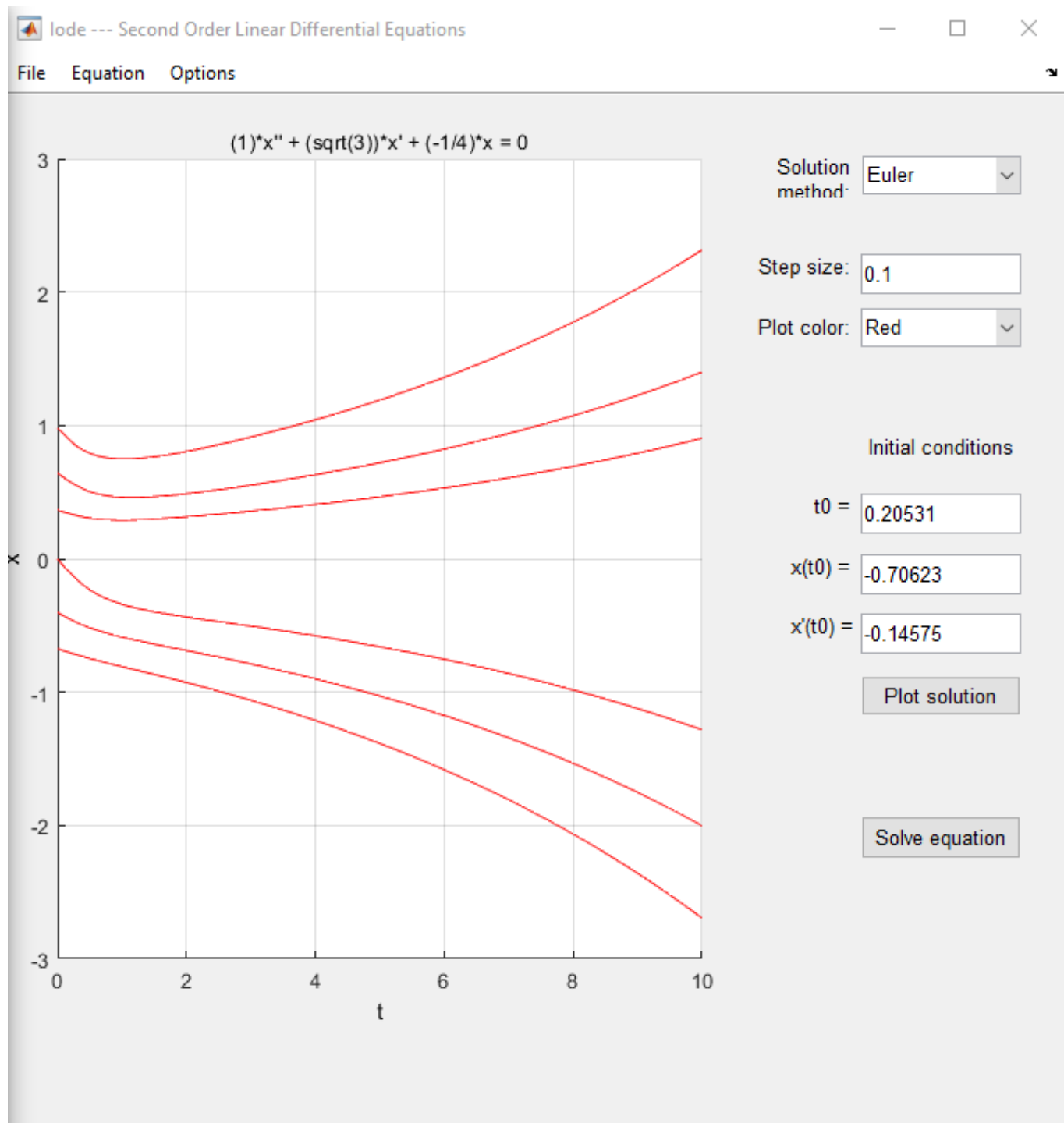
Exercise 2

Consider the ODE:

$$y'' + \sqrt{3} y' - y/4 = 0$$

Repeat (a), (b), (c) from Exercise 1 with this DE.

a)



b) All solutions (100%) grow (no oscillation)

c) The exact solution is $y = c_1 e^{((- \sqrt{3} + 2)/2)t} + c_2 e^{((- \sqrt{3} - 2)/2)t}$. This is justified as the solution will grow for all t , since the power of the exponential terms will always be positive. No oscillation occurs as the solution does not contain any periodic function terms

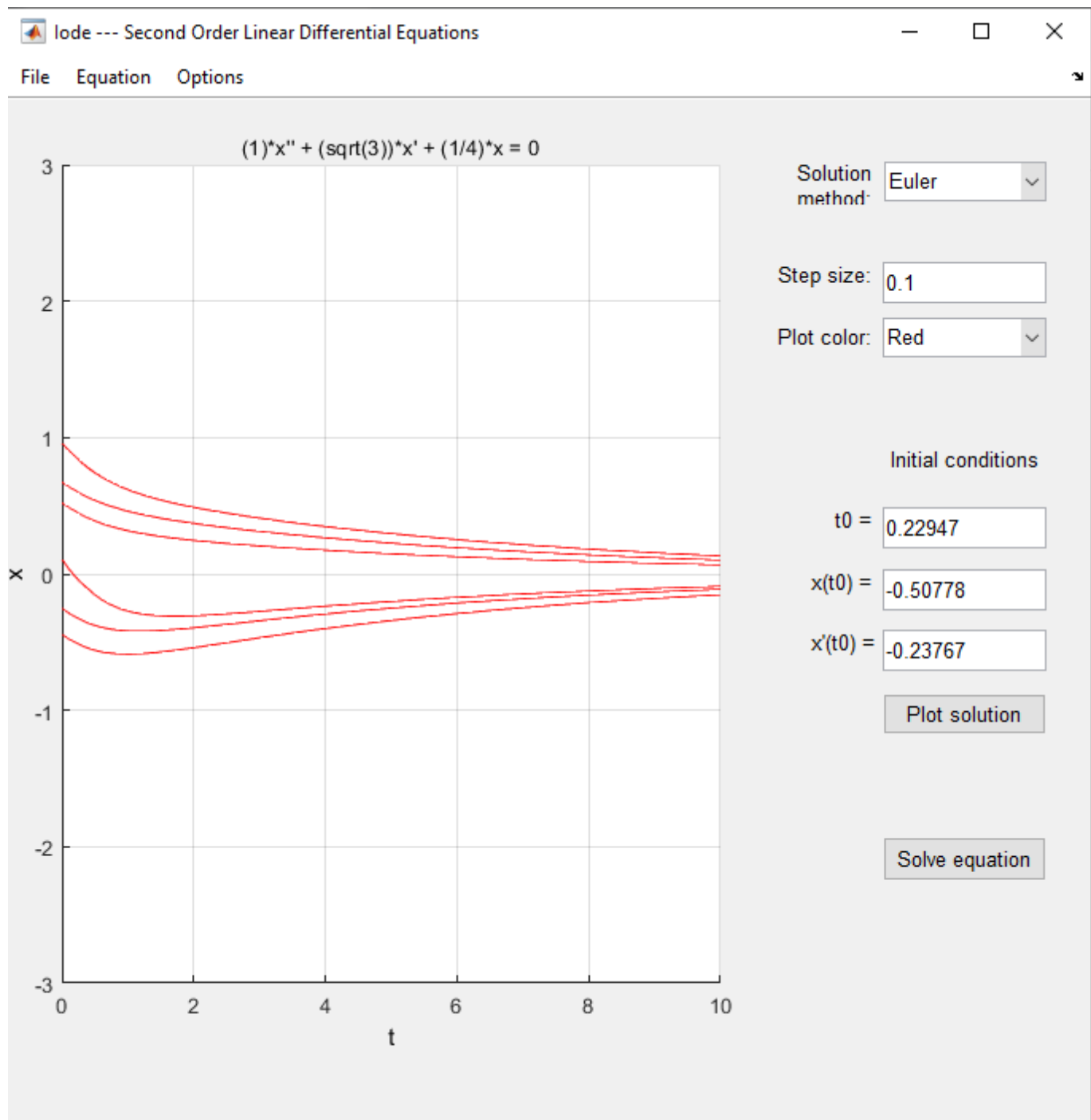
Exercise 3

Consider the ODE:

$$y'' + \sqrt{3} y' + y/4 = 0$$

Repeat (a), (b), (c) from Exercise 1 with this DE.

a)



b) All solutions (100%) decay without oscillation

c) The exact solution is $y = c_1 e^{(-\sqrt{3} + \sqrt{2})t/2} + c_2 e^{(-\sqrt{3} - \sqrt{2})t/2}$. There will be no oscillation as there are no periodic terms and the solution is always decaying as the exponential terms are negative.

Example

Consider the ODE:

$$y'' + 2y' + 10y = 0$$

The solution is

$$y(t) = e^{-t} (c_1 \cos(3t) + c_2 \sin(3t))$$

From this, it is easy to see that all solutions decay while oscillating.

Similarly, for the equation

$$y'' - 2y' + 10y = 0$$

The solution is

$$y(t) = e^t (c_3 \cos(3t) + c_4 \sin(3t))$$

which grows while oscillating.

Exercise 4

Consider the fourth-order ODE:

$$y'''' + 2y''' + 6y'' + 2y' + 5y = 0$$

(a) Find the general solution for this problem. You can use MATLAB to find the roots of the characteristic equation numerically with `roots` or symbolically with `solve`.

(b) Predict what percentage of solutions with random initial data will grow, decay, grow while oscillating, and decay while oscillating. Explain.

```
% Below are the roots of the characteristic equation, which give us the
% eigenvalues:

% -1.0000 + 2.0000i
% -1.0000 - 2.0000i
% -0.0000 + 1.0000i
% -0.0000 - 1.0000i

% From the signs, we see that 50% are decaying while oscillating (negative
% real component) and the other 50% are oscillating forever (real component
% is 0)
```

Exercise 5

Objective: Classify equations given the roots of the characteristic equation.

Details: Your answer can consist of just a short sentence, as `grows` or `decays while oscillating`.

Consider a second-order linear constant coefficient homogeneous DE with r_1 and r_2 as roots of the characteristic equation.

Summarize your conclusions about the behaviour of solutions for randomly chosen initial data when.

(a) $0 < r_1 < r_2$

(b) $r_1 < 0 < r_2$

(c) $r_1 < r_2 < 0$

(d) $r_1 = \alpha + \beta i$ and $r_2 = \alpha - \beta i$ and $\alpha < 0$

(e) $r_1 = \alpha + \beta i$ and $r_2 = \alpha - \beta i$ and $\alpha = 0$

(f) $r_1 = \alpha + \beta i$ and $r_2 = \alpha - \beta i$ and $\alpha > 0$

a. Grows b. Saddle c. Decays d. Decays while oscillating e. Centre (oscillating) - Does not grow or decay f. Grows while oscillating

Numerical Methods for Second-Order ODEs

One way to create a numerical method for second-order ODEs is to approximate derivatives with finite differences in the same way of the Euler method.

This means that we approximate the first derivative by:

$$y'(t[n]) \sim (y[n] - y[n-1]) / h$$

and

$$y''(t[n]) \sim (y'(t[n+1]) - y'(t[n])) / h \sim (y[n+1] - 2y[n] + y[n-1]) / (h^2)$$

By writing these approximations into the ODE, we obtain a method to get $y[n+1]$ from the previous two steps $y[n]$ and $y[n-1]$.

The method for approximating solutions is:

1. Start with $y[0]=y_0$

2. Then we need to get $y[1]$, but we can't use the method, because we don't have two iterations $y[0]$ and $y[-1]$ (!!). So we use Euler to get

$$y[1] = y_0 + y_1 h$$

y_1 is the slope given by the initial condition

3. Use the method described above to get $y[n]$ for $n=2, 3, \dots$

Exercise 6

Objective: Write your own second-order ODE solver.

Details: Consider the second-order ODE

$$y'' + p(t) y' + q(t) y = g(t)$$

Write a second-order ODE solver using the method described above.

This m-file should be a function which accepts as variables (t0,tN,y0,y1,h), where t0 and tN are the start and end points of the interval on which to solve the ODE, y0, y1 are the initial conditions of the ODE, and h is the stepsize. You may also want to pass the functions into the ODE the way ode45 does (check MATLAB lab 2). Name the function DE2_<UTORid>.m.

Note: you will need to use a loop to do this exercise.

Exercise 7

Objective: Compare your method with `iode`

Details: Use `iode` to plot the solution of the ODE $y'' + \exp(-t/5) y' + (1 - \exp(-t/5)) y = \sin(2t)$ with the initial conditions $y(0) = 1$, $y'(0) = 0$

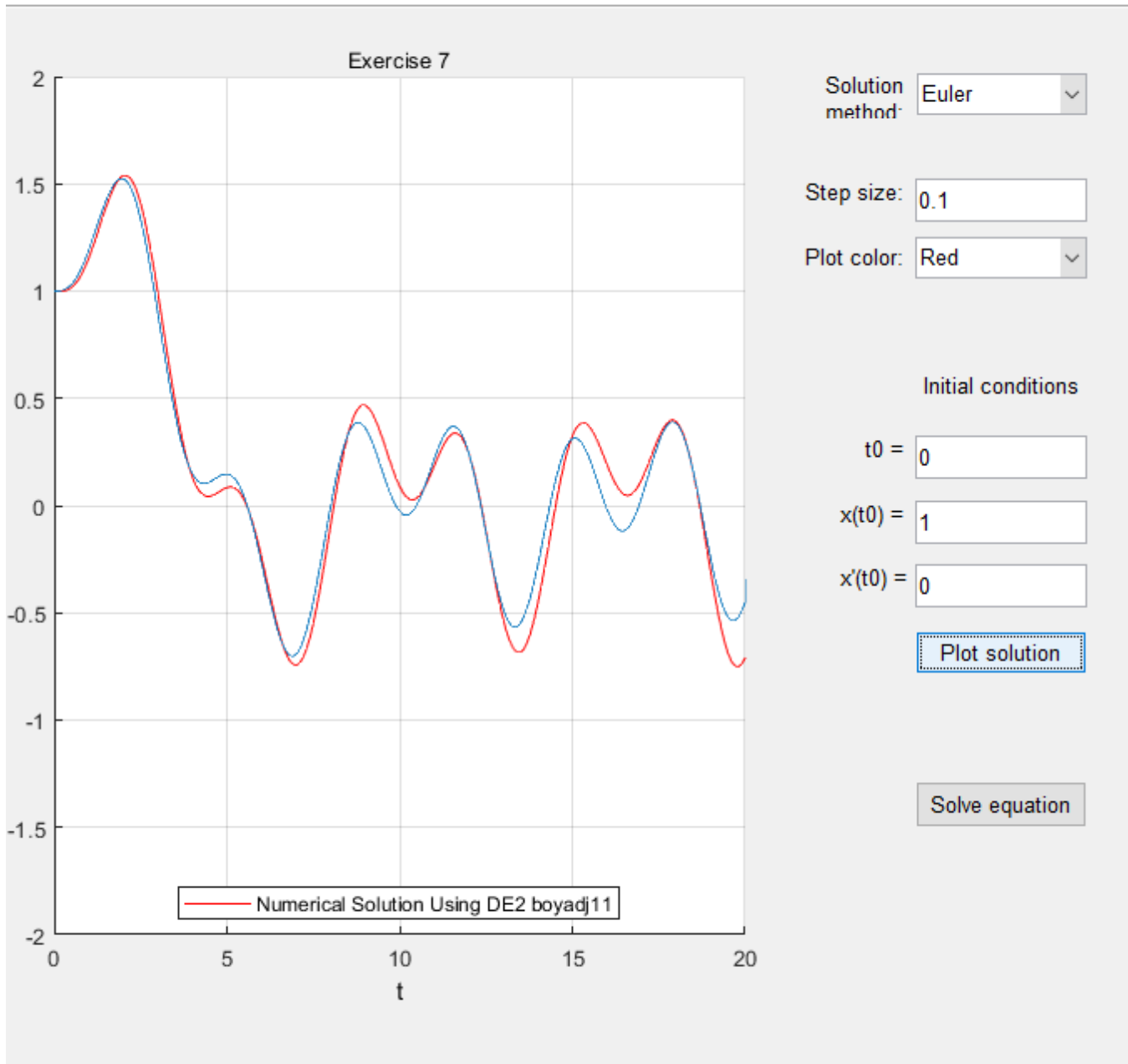
Use the window to $[0, 20] \times [-2, 2]$ Without removing the figure window, plot your solution (in a different colour), which will be plotted in the same graph.

Comment on any major differences, or the lack thereof.

iode --- Second Order Linear Differential Equations



File Equation Options



% Below is the code used to plot the DE2 solver on the iode graph. This was
 % entered in the command window to plot on the same graph

% inline function

```
f = @(t, y, dy) - exp(-t/5)*dy - (1-exp(-t/5))*y + sin(2*t);
```

% numerical solution using DE2_boyadj11 solver

```
[t, y] = DE2_boyadj11(f, 0, 20, 1, 0, 0.01);
```

% plotting on iode plot

```
plot(t, y);
```

```
title('Exercise 7');
```

```
xlabel('t');
```

```
ylabel('y');
```

```
legend('Numerical Solution Using DE2 boyadj11', 'location', 'best');
```

```
% The DE2 solver is nearly identical to the iode solver except that  
% it is a little shifted to the right  
%
```
