Carnegie Mellon University

Department of Electrical and Computer Engineering

18-200 **Fall 2000**

Mathematical Foundations of Electrical Engineering

Problem Set 3

Due: Tuesday, October 10

Do the following problems from these Problem Set sections of Kreyszig, chapter 2. For each problem, you will find a solution y(x) for the given 2nd order constant-coefficient linear homogenous differential equation. You should do the following three things:

a) Solve the equation for y(x), using the differential equation and initial conditions.
b) Check your result, by differentiating your result for y(x) and substituting it back into the differential equation and showing that it solves the equation. Check the initial conditions too.
c) Use MATLAB to plot y(x) for the range given with each problem.

Section 2.2 p. 75

1. Problem 11
(plot y(x) between 0 and 2)

2. Problem 12
(plot y(x) between 0 and 2)

3. Problem 13
(plot y(x) between 0 and 2)

4. Problem 14
(plot y(x) between 0 and 2)

Section 2.3 p. 80

1. Problem 15
(plot y(x) between 0 and 2)

2. Problem 17
(plot y(x) between 0 and 2)

3. Problem 19
(plot y(x) between 0 and $\pi/2$)

As an example of plotting a function y(x) in MATLAB:

Suppose we'd like to plot $y(x) = e^{-x} \cos(2x)$ for values of x between 0 and π.

```
x = [0:0.01:1]*pi;          (creates a "vector" x of 101 values from 0 to p)
y = exp(-x).*cos(2*x);      (calculates the corresponding values of y(x) for the range of x above.
                                 Be sure to use .* instead of just * to multiply the two functions, exp(-x) and cos(2*x). )
plot(x,y)                   (plots y vs. x)
plotlimits = [0, pi, -1, 1]; (creates a vector of values to set plotting limits)
axis(plotlimits)            (uses the "plotlimits" above to reset the endpoints x and y axes on the plot)
grid                             (adds a grid to the plot)
xlabel('x')                      (puts a label on the x-axis)
ylabel('y')                      (puts a label on the y-axis)
title('example plot')            (puts a title on the plot.)
```

Putting a semicolon (;) after an expression that generates a lot of values causes MATLAB to not echo all of those values to the screen (can be annoying to watch a ton of values stream by on the screen!).