MAT 330: Differential Equations

Module One Template

Complete this template by replacing the bracketed text with the relevant information.

[JoAnn Olney]

[5/9/25]

Plotting Functions

Introduction:

The solution to a differential equation is a function that satisfies the original differential equation. Learning various techniques for solving differential equations is one of the primary goals of this course.

It is often desirable to plot the computed solution to analyze its characteristics or to verify the solution behaves as expected.

In this MATLAB assignment, you'll learn how to define symbolic variables, define symbolic functions, and plot symbolic functions. These plotting skills should be helpful throughout the course.

Review the code and comments provided in the "Examples" section below, and then use this information to complete the problems listed in the "Problems" section.

Make sure to run your code so all relevant computations/results are displayed, delete the Introduction and Examples sections, and then export your work as a PDF file for submission. Your submission needs to contain only the problems that you completed.

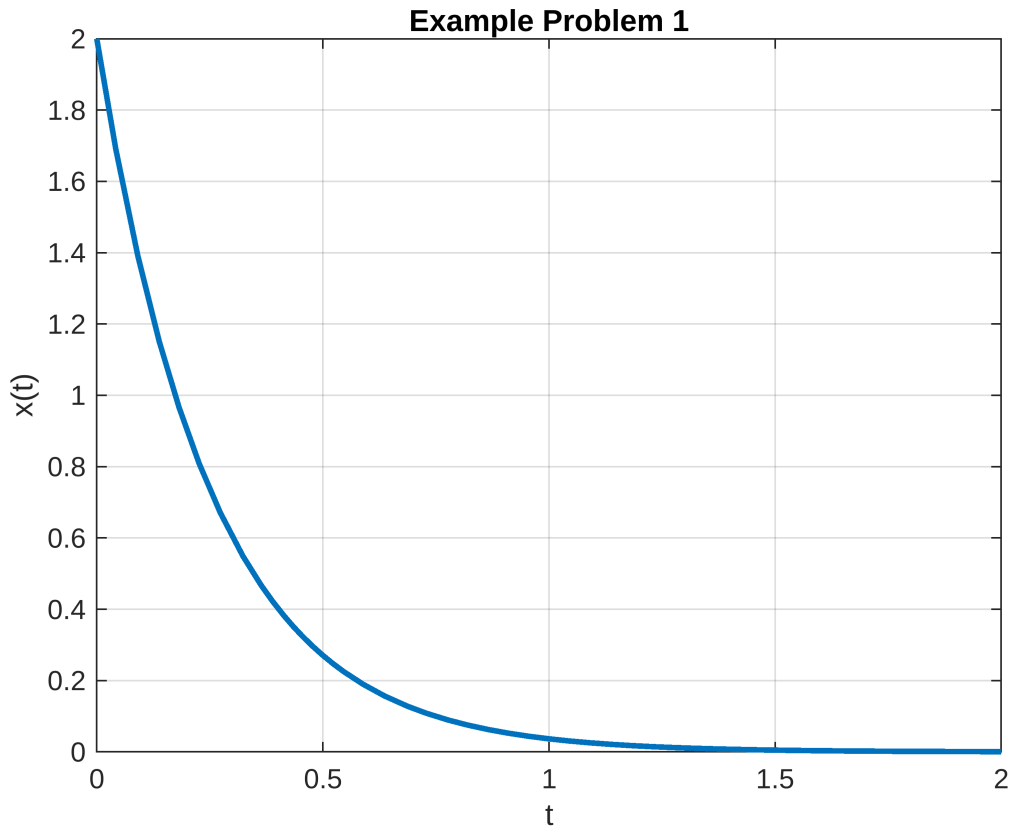
Examples:

```
%Example 1 Code

% The "syms" command is used to define symbolic variables with the MATLAB
% workspace. You can define all symbols in a single list.
syms t A C;

% You can assign symbolic variables a value using the "=" sign.
A = 2;
C = -4;
```

```
% If you define x(t) as function of the variable "t", the quantity x(t)
% will now be a symbolic function within the MATLAB workspace. Note, this
% line is not "terminated" with a semi-colon ";". Terminating with ";"
% keeps the line from outputing to the display and most of the time this is
% desired. However, if you do want to see the value, remove the semi-colon
% to allow it to print out.
x(t) = A*exp(C*t)
```

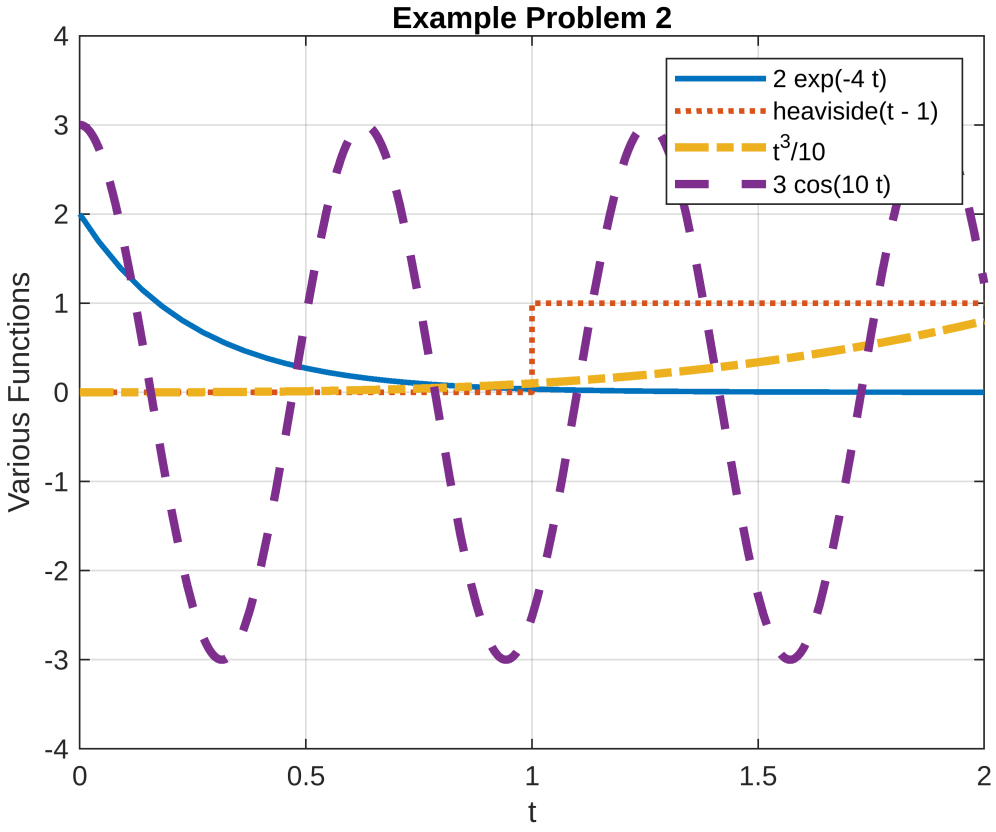
x(t) = $2e^{-4t}$

```
%Other frequently encountered functions include cos(t), sin(t),
%heaviside(t). Powers/polynomials are defined using "^", e.g. t^4, etc.

% The "figure" command starts a new figure/window
figure;

% The "fplot" function plots a symbol function over some interval. The
% second argument [0,2] tells fplot to plot from time t = 0 to t = 2.
% The 3rd argument, '-', tells fplot to make the line solid. Other line
% types include dotted :, dashdot -., and dashed --
% The width of the line is controlled by the optional 'linewidth'
% argument (in this case the linewidth is set to 2).
fplot(x(t),[0,2],'-','linewidth',2);

% It's usually best practice to title your figure and label the axes. The
% xlabel(), ylabel(), and title() functions are used for this.
xlabel('t');
ylabel('x(t)');
title('Example Problem 1');

% Plotting on a grid often improves the look of a plot. The "grid on"
% command will turn on a grid.
grid on;
```




```
% Example 2 Code
% Let's define some additional functions and plot everything in a single
% figure. Any variables or symbols defined above are in the workspace so
% no need to re-define x(t).
x2(t) = heaviside(t-1);  %this is a unit step function u(t)
                         %that turns on a time t = 1.
                         %u(t) = heaviside(t)
x3(t) = 0.1*t^3;
z(t)  = 3*cos(10*t);

% The code below is similar to Example 1, just plotting different functions
% with different line types.
%
% The "hold on" command tells MATLAB to keep all figures on a single plot.
% Without this command, each new plot replaces the previous one.
figure;
fplot(x(t),[0,2],'-','linewidth',2);
hold on;
fplot(x2(t),[0,2],':','linewidth',2);
fplot(x3(t),[0,2],'-.','linewidth',3);
fplot(z(t),[0,2],'--','linewidth',3);
grid on;
xlabel('t');
ylabel('Various Functions');
title('Example Problem 2');
```

```
%the legend function can be used to add a legend to the figure
legend;

%The "ylim()" and "xlim()" functions can be used to manually set the limits
%of the figure. In this case, lets change the y-axis to zoom out a bit.
% set the yaxis to zoom out a bit
ylim([-4 4]);
```



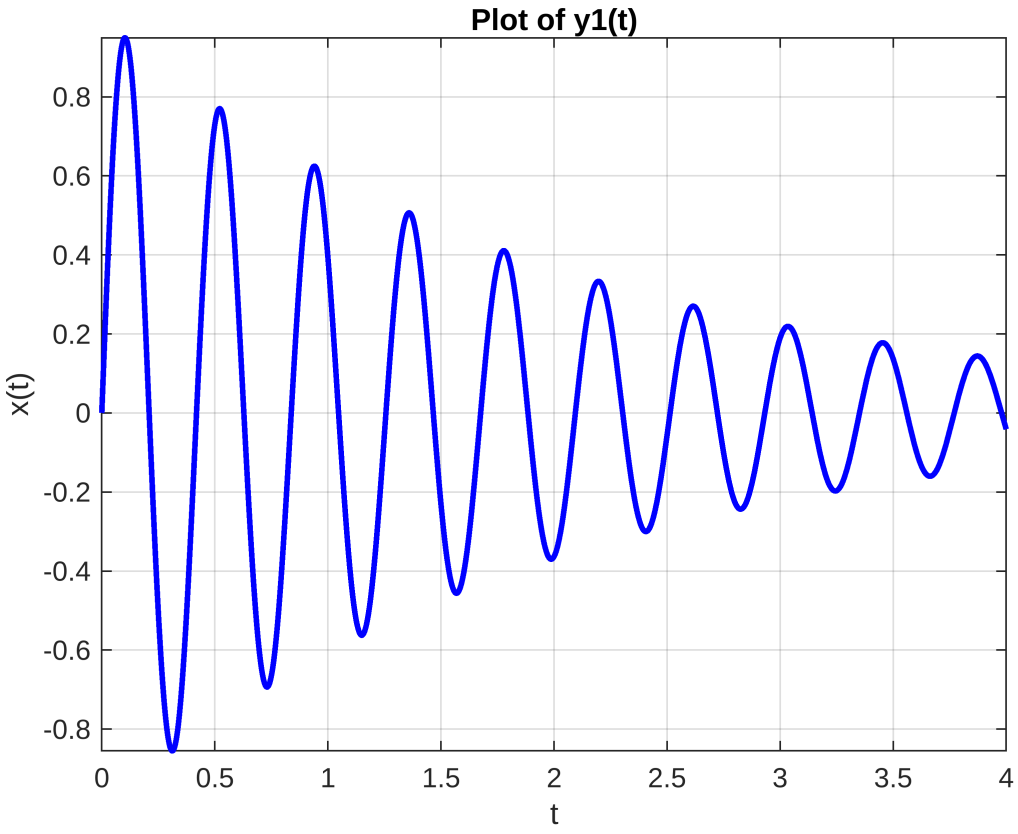

```

```

Problems:

Problem 1: Use MATLAB to plot the function $y_1(t) = e^{-0.5t}\sin(15t)$ for $t = 0$ to 4. Plot the function as a solid line and turn the plotting grid on. Make sure to label both axes and title your figure.

```
%[Insert your MATLAB code and plot here.]
%define variable t
syms t;

%define symbolic function y1(t)
y1 = exp(-0.5 * t) * sin(15 * t);

%create figure
```

```
figure;

%use the fplot to plot the symbolic function over the given range
fplot(y1, [0, 4], 'b-', 'LineWidth', 2);

%label both axes and title the graph
%turn grid on
xlabel('t');
ylabel('x(t)');
title('Plot of y1(t)');
grid on;
```



Problem 2: Use MATLAB to plot the functions $z_1(t) = e^t$, $z_2(t) = 2u(t+1)$, and $z_3(t) = 0.5\sin(7t)$ for $t = -2$ to 2. Plot $z_1(t)$ function as a solid line, $z_2(t)$ as a dotted line, and $z_3(t)$ as a dashed line, and turn the plotting grid on. Make sure to label both axes, title your figure, and turn on the plotting legend. Set the y-axis limits to [-1 8].

```
%[Insert your MATLAB code and plot here.]
%define symbolic t variable
syms t;

%define the symbolic functions z1, z2, z3
```

```
z1 = exp(t);
z2 = 2 * heaviside(t + 1);
z3 = 0.5 * sin(7 * t);

%create the figure
figure;

%plot the symbolic functions over the given range
fplot(z1, [-2, 2], 'b-', 'LineWidth', 2);
hold on; %hold on, dash lines after this
fplot(z2, [-2, 2], 'r:', 'LineWidth', 2);
fplot(z3, [-2, 2], 'g--', 'LineWidth', 2);

%label all axes, title the grid
%turn grid on
grid on;
xlabel('t');
ylabel('Functions');
title('Problem 2 Figure');

%add a legend
legend;

%set yaxis limits
ylim([-1, 8]);
```