MATLAB Help Handout

OPENING MATLAB

To open Matlab, click **Start-> All Programs->Math Programs -> MATLAB R2015a**. Matlab is loading, and may take a few seconds before the *MATLAB Command Window* opens.

Commands can either be typed directly into the command prompt of the *MATLAB Command Window*, or an *M-file* may be used to store the code. To create a new *M-file*, click **New Script,** located at the top of the screen under the Home tab in the File pane. This opens the *Matlab Editor/Debugger*. The code can then be typed in the window that opens. To execute the code, click on **Run** button under the Run pane in the Editor tab. Alternatively, save the *M-File*; then type the name of the *M-File* at the command prompt.

DEFINING AND USING ARRAYS

In the *Matlab Editor/Debugger*, the percent sign ("%") signifies a comment line. The title of the project is written to specify what the code is for. In this example, type:

```
%DEFINING AND USING ARRAYS
```

Define two matrices A and B. There are various ways to denote the end of a row in Matlab. Matrix A uses a line return to denote the end of a row, while matrix B uses semicolons.

```
A=[10 20 30 40 50
   60 70 80 90 10
   20 30 40 50 60
   70 80 90 44 34
   32 65 76 32 11];

B=[90; 80; 70; 60; 50];
```

Be sure to include the semi colon at the end of each number in B. Otherwise you will input a horizontal matrix instead of a vertical one!

The semicolon (;) at the end of each line hides what A and B are. To view A and B when the command is executed, omit the semicolon.

```
%DEFINE C AS A TRANSPOSE
```

The apostrophe (') at the end of Matrix C, transposes the matrix.

```
C= A'
```

The resultant matrix is shown below.

```
C =
    10    60    20    70    32
    20    70    30    80    65
    30    80    40    90    76
    40    90    50    44    32
    50    10    60    34    11
```

```
%MULTIPLYING TWO MATRICES
```

```
D=A*B
```
The resultant matrix is shown below.
```
D =

        9500
       22500
       13000
       23340
       15870


%IF A*X=B, TO SOLVE FOR X:
X=inv(A)*B
```
The resultant matrix is shown below.
```
X =

    2.8977
  -23.0455
   19.3977
   -0.2500
   -1.0000

%GIVES SAME RESULT
Y=A\B
```
Note that that is a **forward** slash not a back slash. The resultant matrix is shown below.
```
Y =

    2.8977
  -23.0455
   19.3977
   -0.2500
   -1.0000
```

Save the m-file in the C: directory by clicking **File-> Save** in the MATLAB editor/debugger toolbar. In the window that opens, type the name of the file. To execute the program, click **Run** in the Run pane of the Editor tab or type the name of the *M-File* at the command prompt in the *MATLAB Command Window*.

SIMPLE PLOTS

Follow the steps described in the *OPENING MATLAB* section to open a new m-file, or add this code to the previous M-File. Comment this code as:

```
%Simple sine wave plot
```

Matlab automatically numbers figures created in numerical order. The user can also define which figure number is associated with the data. Typing figure(1) ensures this set of data plots into figure 1.

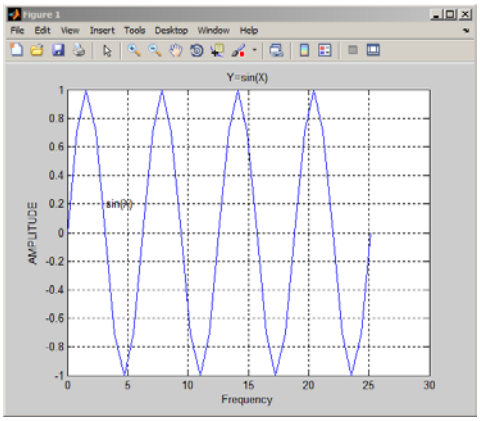
```
figure(1)

%DEFINE X – This sets the interval for which “y” will be plotted;
x=[0*pi:.25*pi:8*pi];

%Define the function y
y=sin(x);
%The plot command is then used to plot the function
plot(x,y);
```

```
%The following commands are “extras” that can be added to the plot.
%Grid places a grid in the background, and text places text at the
%given location.
grid
title('Y = sin(x)');
xlabel('Frequency');
ylabel('Amplitude');
text(3.2, 0.2,'sin(x)');
```

As described in the previous example, to run the program, save the m-file to the C: directory. In the MATLAB Command Window, type the name of the file and hit enter. This will run the program. A window titled “Figure No. 1” opens showing the plot. The completed plot is shown below.



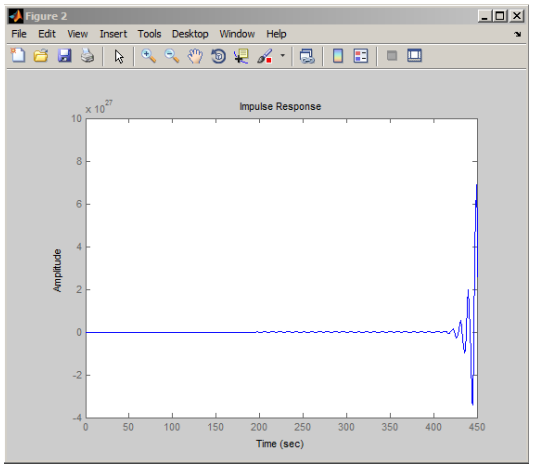
OTHER FUNCTIONS

The impulse response of function shown below is obtained using the “impulse” command.

$$H(z) = \frac{4x^4 + 5x^3 - 7x}{7x^4 + 3x^2 + x}$$

```
%Declare the Numerator in order of MS power
num= [4 5 0 -7 0];
%declare the denominator in MS powers
den= [7 0 3 1 0];
figure(2);
impulse(num,den);
```

The impulse response is shown below.




```
%Bode_Plot
```

Bode plots are a very useful way to represent the gain and phase of a system as a function of frequency. This is referred to as the frequency domain behavior of a system.

The Bode Plot of the following function is found using the Matlab command Bode.

$$H(z) = \frac{1x^6 + 4x^5 + 23x^4 + 8x^3 + 30x^2 + x}{2x^4 + 18x^3 + 3x^2 + 26x}$$

To define the function in terms that Matlab understands, the numerator and denominator are defined with the coefficient of the variables starting with the most significant power.

```
num=[1 4 23 8 30 1 0];
den=[2 18 3 26 0];
```

The command for the bode plot is simply “bode”.

```
bode(num,den)
```

```
The Bode Plot is shown below
```