BME1901 – Introductory Computer Sciences
Laboratory Handout – 1

OBJECTIVES

Learn about,

- Vectors
- Matrices
- Indices of Vectors/Matrices
- Vector/Matrix manipulation
- Built-in functions [ones(), zeros(), length(), size(), sin(), cos()]
- Graphing [plot()] function

TOOLS

All MATLAB variables are multidimensional arrays, no matter what type of data. To create a vector with four elements in a single row, separate the elements with either a comma (,) or a space.

```
>> a = [1 2 3 4]

a =

     1     2     3     4
```

```
>> a = [1, 2, 3, 4]

a =

     1     2     3     4
```

A matrix is a two-dimensional array often used for linear algebra. To create a matrix that has multiple rows, separate the rows with semicolons (;).

```
>> a = [1 2 3; 4 5 6; 7 8 9]

a =

     1     2     3
     4     5     6
     7     8     9
```

Another way to create a matrix is to use a function, such as **ones()**, **zeros()**. For example, create a 5-by-1 column vector of zeros.

```
>> z = zeros(5,1)

z =

     0
     0
     0
     0
     0
```

```
>> o = ones(2,4)

o =

     1     1     1     1
     1     1     1     1
```

MATLAB allows you to process all of the values in a matrix using a single arithmetic operator or function. For example, adding a value to each element of a matrix.

```
>> a = [1 2 3; 4 5 6; 7 8 9]

a =

     1     2     3
     4     5     6
     7     8     9
```

```
>> a + 10

ans =

    11    12    13
    14    15    16
    17    18    19
```

```
>> a * 10

ans =

    10    20    30
    40    50    60
    70    80    90
```

BME1901 – Introductory Computer Sciences
Laboratory Handout – 1

Every variable in MATLAB is an array that can hold many numbers. When you want to access selected elements of an array, use indexing. For example, consider the 4-by-4 magic() square A. The way to refer to a particular element in an array is to specify row (i) and column (j) subscripts [A(i,j)].

```
>> A = magic(4)

A =

    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
>> A(4,2)

ans = 14
```

When you want to manipulate or to change the value of an element in an array, you may again use indexing, however this time on the left hand side of an assignment statement.

```
>> A(1,2) = 17

A =

    16    17     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

To refer to multiple elements of an array, use the colon (:) operator, which allows you to specify a range of the form **start:end**. The colon (:) alone, without start or end values, specifies all of the elements in that dimension.

```
>> A(1:3,1)

ans =

    16
     5
     9
```

```
>> A(3,:)

ans =

     9     7     6    12
```

The colon (:) operator also allows you to create an equally spaced vector of values using the more general form **start:step:end**. If you omit the middle step, as in **start:end**, MATLAB uses the default step value of 1.

```
>> B = 0:10:100

B =

     0    10    20    30    40    50    60    70    80    90   100
```

```
>> C = 0:10

B =

     0     1     2     3     4     5     6     7     8     9    10
```

BME1901 – Introductory Computer Sciences
Laboratory Handout – 1

MATLAB provides a large number of functions that perform computational tasks. Functions are equivalent to subroutines or methods in other programming languages. Functions has two important parts, first is the function name and the second one is its input arguments. To call a function write the function's name and enclose its input arguments in parentheses after it.

L = length(X) returns the length of the array dimension in X. For vectors, the length is simply the number of elements. The length of an empty array is zero.

```
>> v = 5:10

v =

     5     6     7     8     9    10

>> L = length(v)

L = 6
```

[m,n] = size(X) returns a row vector whose elements contain the length of the corresponding dimension of a matrix.

```
>> B = ones(3,5)

B =

     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1

>> [m,n] = size (B)

m = 3

n = 5
```

Y = sin(X) or Y = cos(X) returns the sine of the elements of X in radians. The sin and cos functions operates element-wise on arrays.

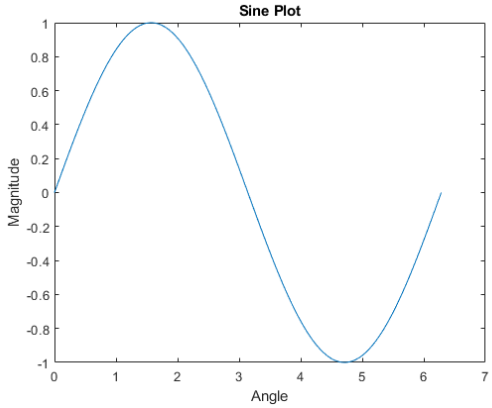
```
>> X = 0:0.5:2

X =

    0    0.5    1    1.5    2

>> Y = sin(X)

Y =

   0   0.4794   0.8415   0.9975   0.9093
```

```
>> X = 0:0.5:2

X =

    0    0.5    1    1.5    2

>> Y = cos(X)

Y =

   1   0.8776   0.5403   0.0707  -0.4161
```

plot(X,Y) creates a 2-D line plot of the data in Y versus the corresponding values in X.

Example: Create x as a vector of linearly spaced values between 0 and 2π. Use an increment of π/100 between the values. Create y as sine values of x. Create a line plot of the data.

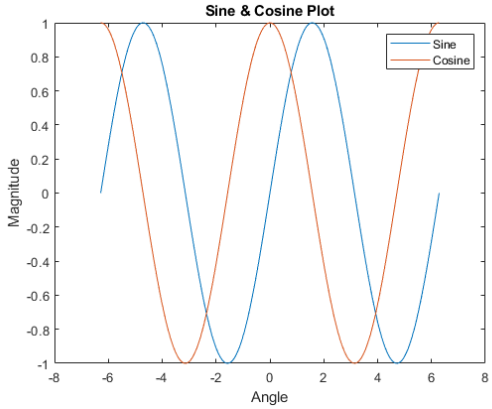
```
>> x = 0:pi/100:2*pi;
>> y = sin(x);
>> plot(x,y)
>> title('Sine Plot')
>> xlabel('Angle')
>> ylabel('Magnitude')
```



Example: Create x as a vector of linearly spaced values between -2π and 2π. Use an increment of π/100 between the values. Create y1 and y2 as sine and cosine values of x. Create a line plot of both sets of data.

```
>> x = -2*pi:pi/100:2*pi;
>> y1 = sin(x);
>> y2 = cos(x);
>> plot(x,y1,x,y2)
>> title('Sine & Cosine Plot')
>> xlabel('Angle')
>> ylabel('Magnitude')
>> legend('Sine','Cosine')
```

```
>> x = -2*pi:pi/100:2*pi;
>> y1 = sin(x);
>> y2 = cos(x);
>> plot(x,y1)
>> hold on
>> plot(x,y2)
>> title('Sine & Cosine Plot')
>> xlabel('Angle')
>> ylabel('Magnitude')
>> legend('Sine','Cosine')
```



BME1901 – Introductory Computer Sciences
Laboratory Handout – 1

PROBLEMS

1. First find the result of the following statements by hand. Then write them to command window to check your results.
    a. 1-6+2*5+8/2^3-3
    b. (9/(3-2))^(1/2)*(6+4)

2. Create a 3x3 matrix using "A = magic(3)" in command window. Then do the following.
    a. Take the first and second rows of matrix A and add them together to create vector B.
    b. Take the first and third column of the matrix A and multiply them element-wise to create vector C (Hint: if you only use (*) you will encounter an error. Search help section of MATLAB for "element-wise multiplication").
    c. Find matrix multiplications of B*C and C*B (additionally check the sizes of both B and C matrices to notice that they are 1x3 and 3x1, respectively).

3. Create a time vector t from 0 to 10 seconds with 0.01 second steps. Create an acceleration of a at 1.1334 m/s^2 and an initial velocity of v0 at 8.5487 m/s. Using the formula for velocity $v = v_0 + a \times t$ calculate velocity v for all t. Plot the resulting velocity against time. Then using disp() function show velocity values for each second (0, 1, 2, …, 10) on command window.

4. Create a time vector t from 0 to 10 seconds with 0.1 second steps, then create a frequency of f at 0.5 Hz. Create magnitudes of A and B at 2 and 5, respectively. Additionally, crate a phase of p at 0.2π.
    a. Write the statement to find signal y1 such that it is equal to "A sin(2π f t)".
    b. Write the statement to find signal y2 such that it is equal to "B sin(2π f t + p)".
    c. Plot the signals y1 and y2 on the same graph against time.
    d. Name the graph title as "Problem 3 Sine Graph".
    e. Name the y axis label as "Magnitude".
    f. Name the x axis label as "Time".
    g. Create legend for both signals with names "Sine 1" and "Sine 2".

HINTS FOR POOL PROBLEMS

You may study the following functions using MATLAB's Help for future questions.

imread(): Read image from graphics file.

imshow(): Display image.

rgb2gray(): Convert RGB image or colormap to grayscale.

A grayscale image is just a matrix of brightness values corresponding to each pixel.