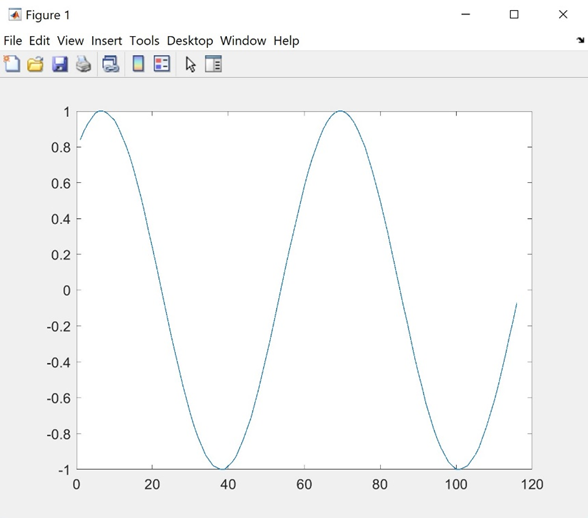
**BE2080 Week1 Assignment**

Download the text-formatted data (HW1.txt) and plot its data interactively.

1. Save the enclosed HW1.txt to your computer.
2. Use the interactive ‘Import Data’ option from your Matlab software to read in this data. Please note that the default “Output Type” is set to Table in newer Matlab. You have to change it by selecting “Numeric Matrix” in order to import the file as a matrix.
3. Use the interactive plot option from the menu bar to plot the data.

**Submission:** copy and paste the figure you created in your homework for submission

****

2) Think about what the results would be for the following expressions, and then type them in Matlab to verify your answers.

>> 5\*50/5

ans =

50

>> 5+3^2

ans =

14

>> 4^2\*2

ans =

32

>> 3+5\15

ans =

6

>> 4-2\*3

ans =

-2

3) Find and write down the MATLAB expressions for the following calculations:

π1.2

tan(50°)

>> sqrt(64)

ans =

8

>> pi^1.2

ans =

3.9498

>> tand(50)

ans =

1.1918

4) In the ASCII character encoding, the letters of the alphabet are in order: ‘a’ comes before ‘b’ and also ‘A’ comes before ‘B’. However, which comes first - lower or uppercase letters?

Uppercase comes first

>> numequiv = double('a')

numequiv =

97

>> numequiv = double('A')

numequiv =

65

5) Use **help elfun** to understand the meaning and syntax of the following rounding functions,or experiment in Matlab to answer the following questions:

* Is **fix(3.5)** the same as **floor(3.5)**? Yes; 3
* Is **fix(3.2)** the same as **floor(3.2)**? Yes; 3
* Is **fix(-3.2)** the same as **floor(-3.2)**? No; -3 and -4
* Is **fix(-3.2)** the same as **ceil(-3.2)**? Yes; -3

>> fix(3.5)

ans =

3

>> floor(3.5)

ans =

3

>> fix(3.2)

ans =

3

>> floor(3.2)

ans =

3

>> fix(-3.2)

ans =

-3

>> floor(-3.2)

ans =

-4

>> fix(-3.2)

ans =

-3

>> ceil(-3.2)

ans =

-3

6) An engineer has determined that the cost C of a containment tank will be based on the radius *r* of the tank:

C = 

Using Matlab, create a variable for the radius (r), assign a value of 15 to it, and then calculate for the cost.

>> r = 15;

>> c = 32430/r + 428\*pi\*r;

>> c

c =

2.2331e+04

7) Using the **colon operator**, create the following three row vectors

2 3 4 5 6 7

1.7 1.5 1.3 1.1

9 6 3 0

>> vec1 = 2:7

vec1 =

2 3 4 5 6 7

>> vec2 = [1.7:-0.2:1.1]

vec2 =

1.7000 1.5000 1.3000 1.1000

>> vec3 = [9:-3:0]

vec3 =

9 6 3 0

\*8) Using two different methods to create a column vector that has the values -2 to 2 in steps of 0.5.

>> vec = [-2:0.5:2]'

vec =

-2.0000

-1.5000

-1.0000

-0.5000

0

0.5000

1.0000

1.5000

2.0000

>> vec = [-2; -1.5; -1; -0.5; 0; 0.5; 1; 1.5; 2]

vec =

-2.0000

-1.5000

-1.0000

-0.5000

0

0.5000

1.0000

1.5000

2.0000

9) Generate a 4 x 3 matrix variable *mat* with all ones. Replace the 2nd column with 1:4. Replace the third column with all 3s.

>> mat = ones(4, 3)

mat =

1 1 1

1 1 1

1 1 1

1 1 1

>> mat(:, 2) = 1:4

mat =

1 1 1

1 2 1

1 3 1

1 4 1

>> mat(:, 3) = 3

mat =

1 1 3

1 2 3

1 3 3

1 4 3

10) Write a Matlab expression that extracts only the “odd-positioned” elements in a vector (i.e. the 1st, 3rd, 5th, …, elements), **regardless of the length of the vector**. Test your expression on vectors that have both an odd and even number of elements.

For example, your expression extracts [10 30 50] from the vector [10 20 30 40 50] or the vector [10 20 30 40 50 60].

>> vec = [10 20 30 40 50]

vec =

10 20 30 40 50

>> vec(1:2:length(vec)) = [] **\* vec(1:2:end) = []**

vec =

20 40

>> vec = [10 20 30 40 50 60]

vec =

10 20 30 40 50 60

>> vec(1:2:length(vec)) = []

vec =

20 40 60

11) Create a *3 x 5* matrix of random numbers, each number is a floating number in the inclusive range from -10 to 10. Perform each of the following:

* Find the maximum value in each column.
* Find the maximum value in each row.
* Find the maximum value in the entire matrix.

>> mat = -10 + (10+10)\*rand(3, 5)

mat =

6.2945 8.2675 -4.4300 9.2978 9.1433

8.1158 2.6472 0.9376 -6.8477 -0.2925

-7.4603 -8.0492 9.1501 9.4119 6.0056

>> max(mat)

ans =

8.1158 8.2675 9.1501 9.4119 9.1433

>> max(mat')

ans =

9.2978 8.1158 9.4119

>> max(max(mat))

ans =

9.4119

\*12) A vector *v* stores for several employees of the Green Fuel Cells Corporation their hours worked one week followed for each by the hourly pay rate. For example, if the variable stores

*>> v =* [33.0 10.5 40.0 18.0 20.0 7.5 28.0 12.5]

that means the first employee worked 33 hours at $10.50 per hour, the second worked 40 hours at $18 an hour, and so on.

Write Matlab expressions to: (1) assign this vector to a variable as shown in the example above; (2) create two more vectors from it, one that stores the hours worked and another that stores the hourly rates; (3) use the array multiplication operator to create a new vector, storing in the new vector the total pay for every employee.

>> v = [33.0 10.5 40.0 18.0 20.0 7.5 28.0 12.5];

>> employee\_stats = v;

>> hours\_worked = employee\_stats([1 3 5 7]); **\*hours\_worked = employee\_stats(1:2:end)**

>> hourly\_rates = employee\_stats([2 4 6 8]); **\*hourly\_rates = employee\_stats(2:2:end)**

>> employee\_total\_pay = hours\_worked.\*hourly\_rates;

>> employee\_total\_pay

employee\_total\_pay =

346.5000 720.0000 150.0000 350.0000