Principles of Mathematics using MATLAB

.Manzur Ashraf*
Department of CSE
East West University

1. Introduction:

Mathworks Inc. designed wonderful software MATLAB to support research and industrial activities in different areas of Engineering and Technology. Technology that is evolving and reaching to the highest extent is approaching more towards efficient initialization and learning process. Moreover, programming from the very root is making the whole process of research as well as effective learning a bit of obscure. "Reusable modules" are helping to make the barrier dismissive. Instead of this, critical time-consuming solution is not caring forth as soon as it is needed. Since, generalization approach of 3G languages can't support in wider prospective of specialized technology or engineering field.

We need two goals in the perspective:

a) Complete visualization capability and clear-cut idea of all the details.

b) Reusability and creativity of modules targeting the significant details of technical areas.

Supporting our two goals, MATLAB has approached to the technical world with its enchanting flavor of interesting visualizations and full phase 3G supportability. Current version of MATLAB (6.5 when this writing is going on) comprises more then 30 toolboxes different Electrical and Electronics, Civil Engineering, Mechanical Engineering, Aerospace Science, Telecommunication fields etc. There are lots of chance to make any reusable components in MATLAB to turn it to any specialized field of engineering and technology. It's not the end, Biological Science, Geo Science, or even fine arts can contribute to MATLAB with giving reusable components and exploring it to the research and appliances.

[1] * Corresponding author: manzur_bd@yahoo.com; Tel:018219755

2. MATLAB in Computational Computer Science:

After an inspiring introduction and giving a brief capability in different fields of technology, I would like to confine my writing to computation related to Computer Science. Readers can install MATLAB 6.5 (version 7 is also available) or any previous version to see the effects of the codes below.

A. Finding out roots:

MATLAB can take input of n polynomial or function as the following notation

$$a_n x^n + \ldots\ldots\ldots + a_2 x^2 + ax + b = 0$$

where a and b are coefficients and constants as well. Suppose, I would like to take the following function as input:

$$3x^3 + x^2 - 2x + 5 = 0$$

In command prompt user should type as follow

```
>> F = [3 1 –2 5];
```

when F is any variable (like 3G languages). To find out root, we can call reusable module like "root ()"

```
>> root (F);
```

The above function will find out all the real and imaginary roots of the polynomial.

B. Plotting a Graph:

We can take (Case 1) input of discrete value of independent (say x) and dependent (say y) variables or a function of independent variable (Case 2) denoting a depending variable (y = f (x)) and plot the curve.

Case 1:

```
>> x = [1 2 3 4];
>> y = [2 4 5 6];
>> plot (x, y);
```

Case 2:

```
>> x = [1 : 0.1 : 10]; // it will increment the value of x as 0.1 from 1 to 10
                          and will make any array of x.
>> y = sin (x); // sin() is another built in function to compute the sine
                  value of x.
```

here Y will calculate all the x(i) values using Sine function and make another array equal to the length of X array.

```
>> plot (x, y);
```

We can customize the X and Y labels or can give title name using 'properties' panel or using command prompt:

```
>> xlabel ('independent variable');
>> ylabel ('dependent variable');
>> title ('plotting an equation');
```

C. Matrix Manipulation:

To take input a row matrix we initiate

```
>> A = [1 2 3];
```

To take input a column matrix we initiate

```
>> B = [2; 3; 4];
```

To take input a matrix of $\begin{vmatrix} 2 & 3 \\ 4 & 5 \end{vmatrix}$ and $\begin{vmatrix} 6 & 7 \\ 9 & 8 \end{vmatrix}$

We will write down:

```
>> C = [2 3 ; 4 5];
>> D = [6 7 ; 9 8];
```

It means every ';' ends a row and goes to the next row. To add, subtract, or multiply two matrix we will initiate

```
>> C + D;
>> C - D;
>> C * D;
```

We can find out determination of a matrix as det(A) and inverse of a matrix as inv(A) command.

D. Solving Linear Equations:

Suppose we are trying to solve the following equations:

$2x + 4y = 6$

$9x + 7y = 8$

We will take input of the above two polynomials as below:

```
>> A = [2  4; 9 7];
>> B = [6 ; 8];
```

Next we can perform any one method given below

```
>> A\B;      // Gauss elimination method
>> inv (A) * B   // Cramer's rule
```

E. Interpolation:

Interpolation is termed as predicting or value of dependent variable upon some value of independent variable. The given value pair (historic data) of {independent, dependent} variables make up an inter relationship among them which helps us to predict value in Interpolation.

For example, some historic data are given as charted:

X	Y
1	21
2	23
3	26
4	27
5	29

Suppose, we want to find out the value of y for the value of x = 3.5 We can perform this query using interpolation. An interesting function "spine" does this operation. At first, we have to take these values of 'x' and 'y' as array.

```
>> x = [1 2 3 4 5];
>> y = [21 23 26 27 29];
>> spine(x, y, 3.5);
```

We will get the predicted value of y for x = 3.5

F. Solving Differentiate Equations:

Differential equation (or rate equation) is playing valuable role in terms of engineering calculations. To find out the original equation from rate equation we need to apply differential equation solving. It requires 'initial value' of dependant variable for independent variable. For example, suppose we want to solve the following differential equation in MATLAB:

. $dy/dx = x - y$ where $y(0) = 1$;

it means when x = 0 (independent variable); y = 1 (dependant variable).

In MATLAB, we will use 'ode23()' or 'ode45()' function to perform this task. In M-file we will first write down the equation in command prompt we will call the above function.

In M-file we will write down,

```
function y = myode(x, y);
.       y = x – y;   // dy/dy = x - y
```

in command prompt, we will call like following:

```
>> [x, y] = ode45('myode', [0 : 0.1 : 10], 1);
```

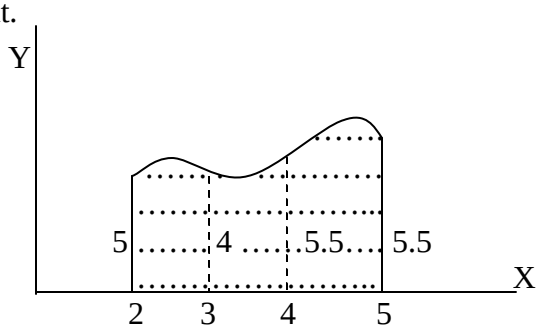
It means, we will find out y for the value of x = 0 to 10 with intermediate increment of 0.1 to x-value. The 3rd parameter (here '1') is the initial value of y. this 'ode45' function will find out all the {x, y} and will store it in [x, y]. We can plot it using plot(x, y) command to visualize the original curve.

G. Integration:

Integration is the process of finding out/ calculating the area/ region. Suppose we are trying to find out an area under the curve $y = x^2 + x$ from starting point x = 0 to ending point x = 5. We will perform $\int(x^2 + x)\,dx$. The procedure is surely based on formula. In numerical method, with the aid of MATLAB we can classify the problem in the cases

1) Discrete and 2) Continuos form. Let's explain these two forms clearly.

Discrete Case: let's consider the following diagram. We are trying to find out the dotted region. Let we partition the total region x-axis by three parts by the points {3, 4}. We calculate the y-distance as {5, 4, 5.5, 5.5}. How can we calculate the whole region? The answer is: we can consider them as trapeziums. Calculate the area of trapeziums. Sum up them and get the result.



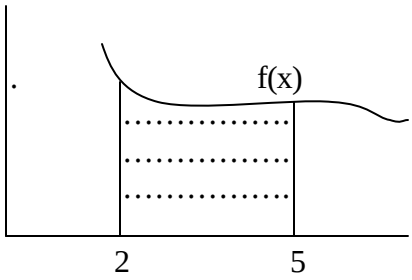
In MATLAB we can perform this using 'trapz()' function.

```
>> X = [2, 3, 4, 5];
>> Y = [5, 4, 5.5, 5.5];
>> trapz (X, Y);
```

Answer: 14.75

So the area we find out is 14.75 square unit.

Continuous Case: whenever we fail to find out the discrete [X, Y] values, we will try to find out the function/ equation under which the region is lying beneath and the range is the start and end x-value. Consider the figure. Suppose we are looking for the dotted area under the function/ equation and in between the range of [2, 5].



Suppose, $f(x) = x^3 + \sin x + e^x$ How can get the area in MATLAB? We will write the function in M-file and call the function using 'quad()' function.

In M-file write down (named "myfile.m")

```
function y = myfile (x);
    y = x^3 + sin (x) + exp (x);
```

in command prompt we have to write:

```
>> result = quad ('myfile', 2, 5);
```

Answer: 292.5753

Thus we get the area of dotted region. Surely in continuous case.

3. Conclusion:

In this article, I tried to highlight the mathematical features that are clearly explained in MATLAB. We can use this useful tool in practical affairs of mathematical Science and Engineering. Lots of calculation difficulties can be resolved using MATLAB. Lots of time and energy to calculate lots of mathematical features can be remitted. Researchers can devote to further thinking and innovating idea, rather then performing or calculating lots of math behind any engineering research. Here, I just mentioned some numerical methods. Furthermore, Digital signal processing, Image analysis, Fourier analysis, Statistics, Communication models visualization, simulations etc lots of interesting tool box are available in MATLAB which help the researchers, educators and engineers in respective technological fields. With world–wide MATLAB contributors (developers of M-file for different activities and functions in Engineering sector) MATLAB is gaining wider prominence in commercial sectors as well as research sectors.