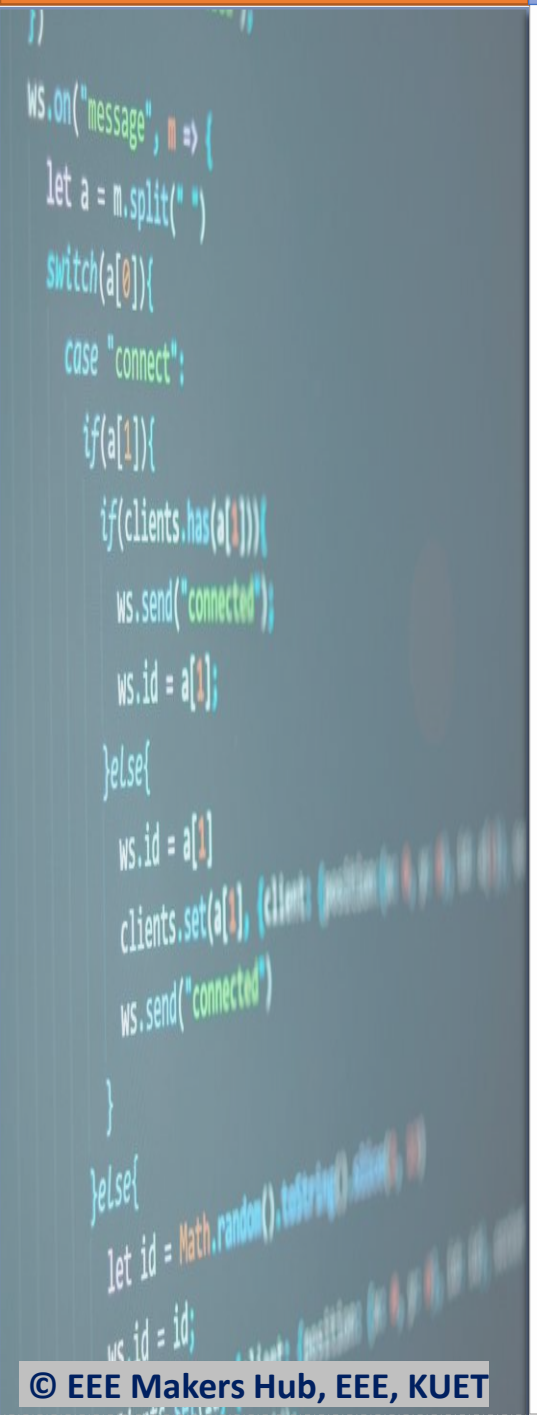
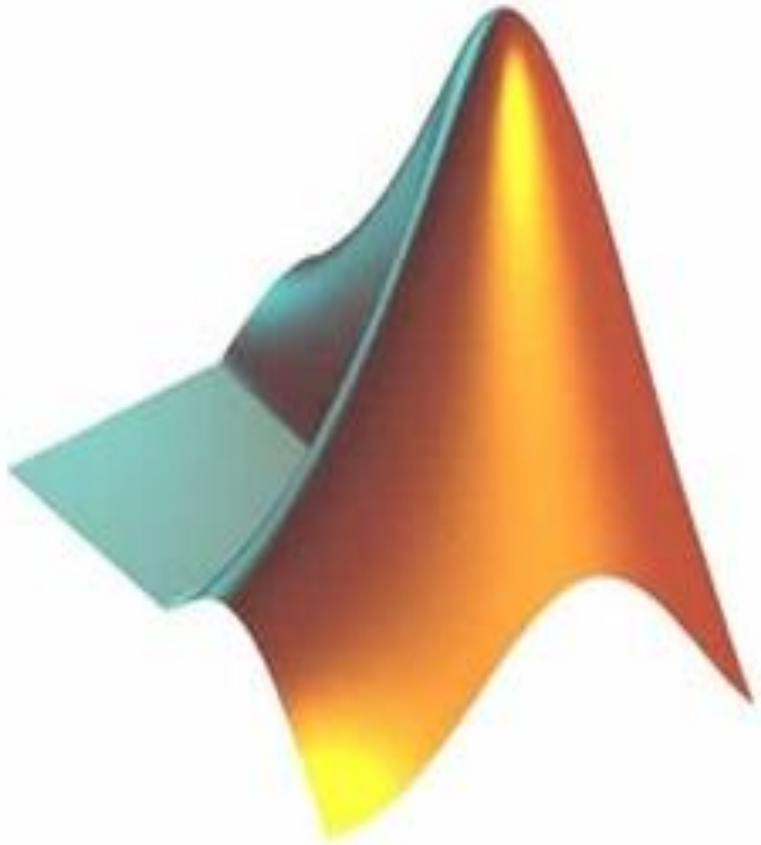




Learning MATLAB

A Case Study Approach

Created by:
Md. Rayid Hasan Mojumder | EEE 2k15 | KUET

Outline

- ❑ Matlab? Why I will even bother to learn it?
- ❑ Target audience
- ❑ Story: Save our 'Jack: the dog'!
- ❑ Intro to the 'Case study approach'
- ❑ Matlab Programming:
  - ✓ Syntax: Variables and Operators
  - ✓ Arrays + Vector + Matrix
  - ✓ Important Commands
  - ✓ M Files
  - ✓ Strings
  - ✓ Data Input and Output
  - ✓ Decision Making
  - ✓ Loops
  - ✓ Functions
  - ✓ 'Jack is back'!
- ❑ Competitive programming
- ❑ Advanced Programming

Matlab? Why I will even bother to learn it?

MATLAB (matrix laboratory) is a fourth-generation high-level programming language and interactive environment for numerical computation, visualization and programming.

Features

- Dealing with Matrices and Arrays
- 2-D and 3-D Plotting and graphics
- Linear Algebra
- Algebraic Equations
- Non-linear Functions
- Statistics
- Data Analysis
- Calculus and Differential Equations
- Numerical Calculations
- Integration
- Transforms
- Curve Fitting
- Various other special functions

Matlab? Why I will even bother to learn it?

"MATLAB® (matrix laboratory) is a programming platform designed specifically for engineers and scientists"

– Mathworks

Application fields

- Signal Processing and Communications
- Image and Video Processing
- Control Systems
- Test and Measurement
- Computational Finance
- Computational Biology

Target audience

Attributes:

WELCOME

- Zero prior experience with Matlab Programming (M. P.)
- want to revise some basics
- ONLY interested in knowing how M. P. works
- I hate code and Matlab, but Hey! Let's waste some time!

If you are:

- One with advance knowledge of M. P.
- Looking for intrique/complex problems and solutions
- Subject specific and competitive programming

Don't waste your VALUABLE time

Save our 'Jack: the dog'!

Storyline:

Jack: the dog had been kidnapped by a terrorist group from the KUET campus yesterday when it was spending some quality time in the lonesome central field. This morning the 'Makers Hub team' has received an anonymous letter that prescribes a direction to locate the exact position of the terrorist gang.

Only Matlab Programmers can save Jack!

To save Jack one needs to follow the following steps:

1) Print the statement 'Jack is missing' to spread the news to all kuetians.
2) Ask honorable V.C. sir for permission. Remember sir will only permit if the dog is a 'good dog' and it's age is more than 2 years.
3) There is a maze with 2 different routes that leads to the cave of the terrorist. Remember the cave is 'red colored', and if there is a 'blue colored' cave you will have to ignore it.
4) Finally, you will have to build a testimony so that each time any kuetian gives his 'KUET roll no' and 'batch name' the terrorist will say 'We are the Worst. Forgive us!'

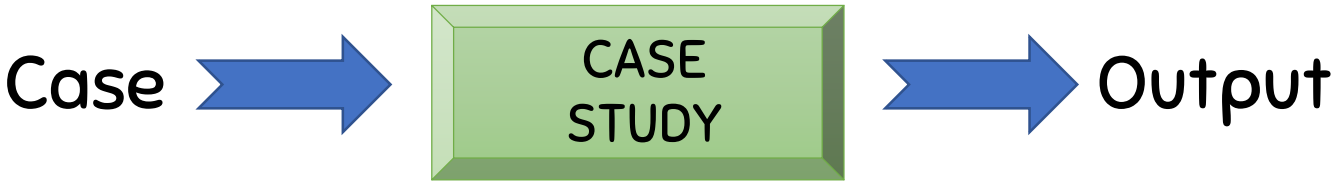
‘Case study approach’

- Guys! “I have just executed ‘for loop’ with 10 conditions within it.”

VS.

- Guys! “I have just executed ‘for loop’ with 10 conditions within it. And it can solely be used for calculating your gpa from your subject scores!”

Each coding exercise, however small let it be, should have a definitive purpose
It should be your nano project



Syntax: Variables and Operators

No need to define 'Variable data type (whether int or float or others)' !

Syntax:

```
var1  +  var2     ! Addition of var1 and var2
var1  >  var2     ! var2 is greater than var2
var1 <=  var2     ! var1 is less than or equal to var2
var1  /  var2     ! Left division: divide var1 by var2
var1  \  var2     ! Right division: divide var2 by var1
var1 ~=  var2     ! Var1 is not equal to var2
```

Operators

```
var4= (var2 * var3) / var5    !assign & show value
var4= (var2 * var3) / var5;   !assign & hide value
```

Semicolon to hide value

More on the topic:
https://www.tutorialspoint.com/matlab/matlab_syntax.htm
https://www.tutorialspoint.com/matlab/matlab_variables.htm
https://www.tutorialspoint.com/matlab/matlab_operators.htm

Arrays + Vector + Matrix

All variables of all data types in MATLAB are multidimensional arrays

1D Array = Vector → 2D Array = 2D Matrix → 3D Array = 3D Matrix

- All arrays are assigned by [ ] square brackets
- Rows are separated by comma (,)
- Columns are separated by semicolon (;)
- For a X array, X(p) refers to the p th element of X array
- For a X matrix, X(r,c) refers to the matrix element which is in the r th row and c th column
- For a X matrix, X([p : r], d) refers to the d th column of the row p to r
- If X, Y and Z are 3 arrays with similar dimensions, then
  D= [X Y Z] will concatenate all the three to one D matrix

More on the topic:
https://www.tutorialspoint.com/matlab/matlab_arrays.htm
https://www.tutorialspoint.com/matlab/matlab_matrics.htm
https://www.tutorialspoint.com/matlab/matlab_vectors.htm

Arrays + Vector + Matrix

Matrix operation:

```
Mat1 + Mat2          ! Matrix addition
Mat1 * Mat2          ! Matrix multiplication
( . ) refers to element by element
Mat1 .* Mat2         ! Element by element multiplication of Mat1 and Mat2
Mat'                 ! ( ' ) refers to transpose (row <-> column) of the matrix
```

Matrix function:

```
det(mat3)            ! Determinant of mat3
inv(mat3)            ! Inverse of mat3
zeros(k)             ! Creates k by k matrix with each element being 0
ones(k, v)           ! Creates k by v matrix with each element being 1
eye(k)               ! Creates k by k identity matrix
Magic(k)             ! Creates k by k magic matrix
rand(k,v)            ! Creates k by v matrix with random numbers betn 0 to 1
```

Important Commands

Commands:

clc	!Clears command window
clear	!Removes variables from memory/workspace
format short	!Four decimal digits (default)
format long	!16 decimal digits
pwd	!Displays current directory
what	!Lists all MATLAB files in the current directory
tic.....toc	!displays the execution time of code within tik and tok
cat	!Concatenates arrays

More on the topic:
https://www.tutorialspoint.com/matlab/matlab_commands.htm

M Files

Create .M file:

- Matlab uses .m extension for scripting (automates the execution of tasks). With one click run multiple commnad

Two ways:

1. Manually using Matlab IDE

   Matlab editor widow –> Open untitle file –> save as –> your targeted directory

2. From command window

   Command window:

```
edit                !will open editor window
edit newfile.m      !will create newfile.m file
mkdir mp            !will create a new mp folder
```

Strings

String syntax:

- String is assigned by (‘ ‘)

```
My_string= ‘This is my string’
length(My_string)

My_string(p)
strcat(a , b)
strcat(‘Hello’ , My_string)
strcmp(str1 , str2)

lower(str1)
upper(str2)
```

```
! Reads a string
! Will print no of character with spaces in the
! string
! Will display p th element of the string
! Will add string a with string b
! Will add ‘Hello’ with My_string
! Will compare str1 and str2, if two are same
! Then returns 1, if not same returns 0
! Convert string to lower
! Convert string to upper
```

More on the topic:
https://www.tutorialspoint.com/matlab/matlab_strings.htm

Data Input and Output

Data input:

```
x = input(prompt)          !shows prompt and wait for a input
str = input(prompt,'s')    !shows prompt and wait for a input and
                           !takes input as a string
```

Data imput:

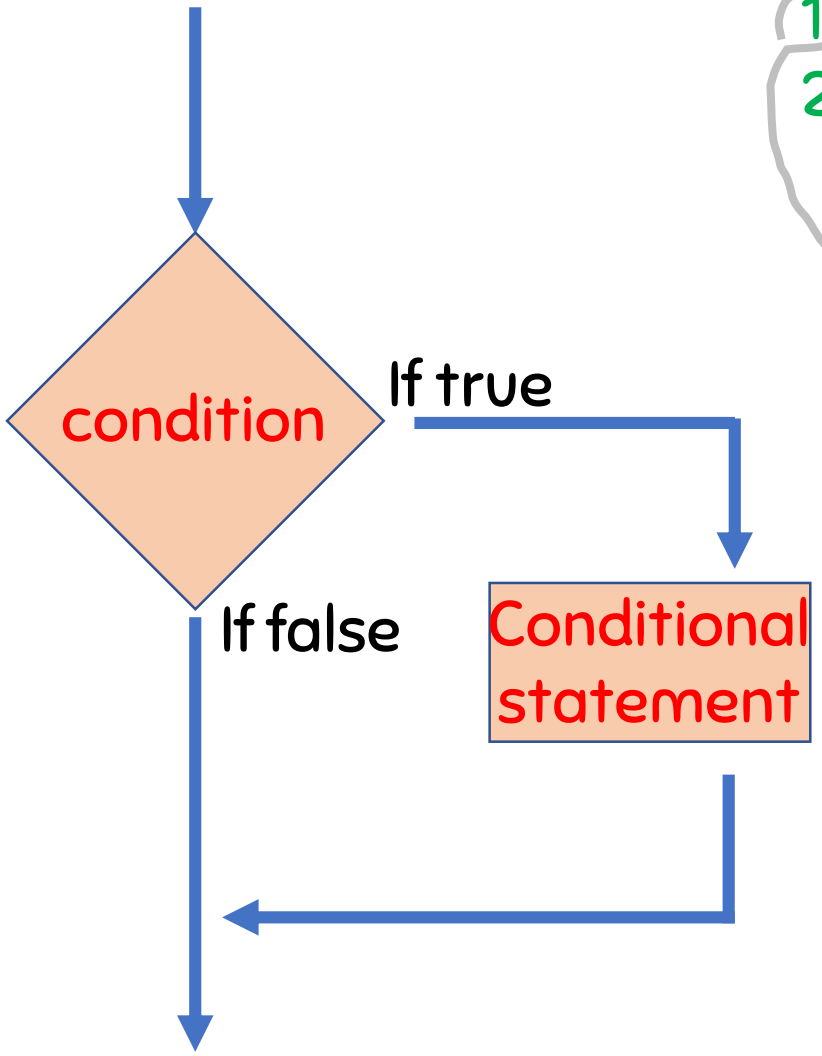
```
Data=xlsread('name.xls')   !imports data from name.xls (Excel data)
load count.dat             !text file saved as .dat extenson. Matlab
                           !reads this and save the file in a variable
                           !named same as filename (here, 'count')
```

Data output:

```
num_array = [ 1 2 3 4 ; 4 5 6 7; 7 8 9 0];
save array_data1.out num_array –ascii; ! ASCII data file created
type array_data1.out
```

Decision Making

Decision:



2 ways:

1. If ...elseif....else.....end
2. switch...case...otherwise....end

```
if <condition 1>
   <statement(s)>
elseif <condition 2>
   <statement(s)>
elseif <condition 3>
   <statement(s)>
....
....
else
   <statement(s)>
end
```

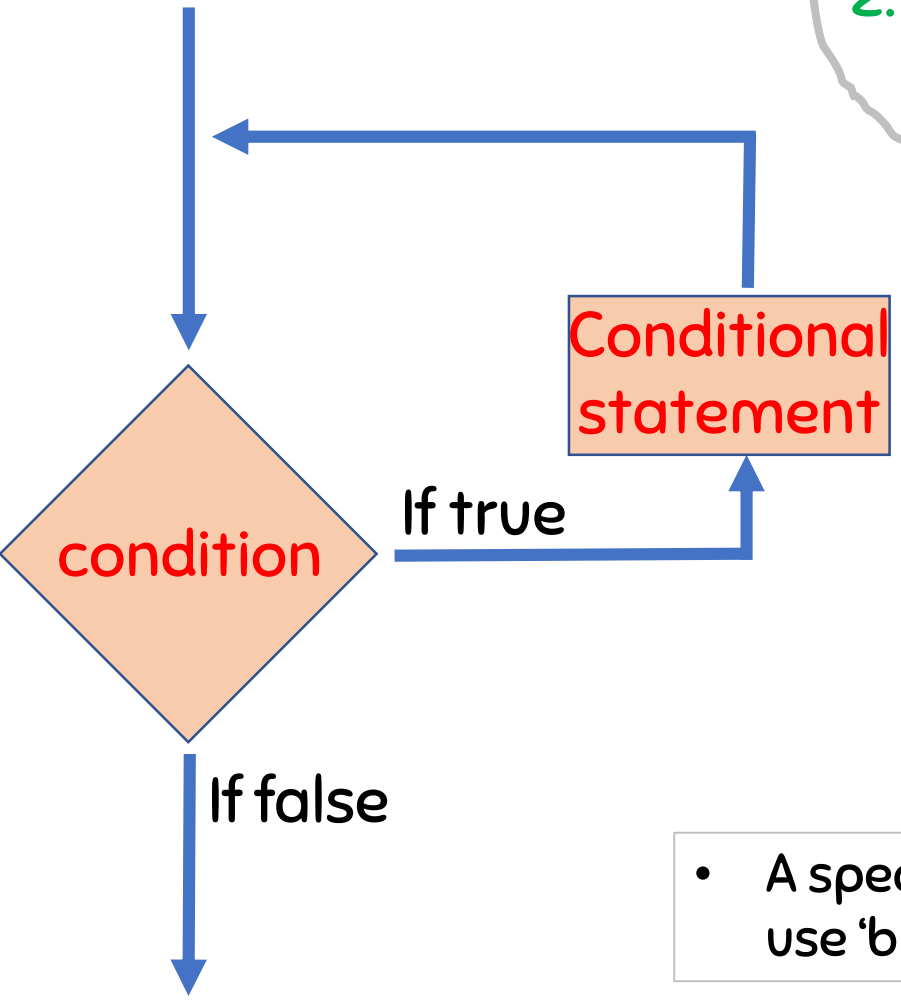
```
switch <switch_expression>
  case <case_expression>
    <statements>
  case <case_expression>
    <statements>
    ...
    ...
  otherwise
    <statements>
end
```

Loops

Iterating:

2 ways:

1. for loop
2. while loop




```
for <loop statement>
  <program statements>
  ...
  ...
end
```

```
while (loop statement)
  <program statements>
  ...
  ...
end
```

<loop statement>

- start_value : step_size : end_value
- conditional

<break statement>

- Breaks the loop

<continue statement>

- Ignores that iteration

- A specific conditional (if.....end) part is added to use 'break' or 'continue' operation

Functions

Functions:

3 types:

1. Built in function : sum(a , b)
2. Anonymous/inline fuction: f = @(arglist)expression
3. User defined function:

```
function [out1,..., outN] = myfun(in1, ..., inN)
```

Returned output

Given input

User defined function name

Call function: myfun(in1, ..., inN)

Save our 'Jack: the dog'!

Storyline:

Jack: the dog had been kidnapped by a terrorist group from the KUET campus yesterday when it was spending some quality time in the lonesome central field. This morning the 'Makers Hub team' has received an anonymous letter that prescribes a direction to locate the exact position of the terrorist gang.

Only Matlab Programmers can save Jack!

To save Jack one needs to follow the following steps:

1) Print the statement 'Jack is missing' to spread the news to all kuetians.
2) Ask honorable V.C. sir for permission. Remember sir will only permit if the dog is a 'good dog' and it's age is more than 2 years.
3) There is a maze with 2 different routes that leads to the cave of the terrorist. Remember the cave is 'red colored', and if there is a 'blue colored' cave you will have to ignore it.
4) Finally, you will have to build a testimony so that each time any kuetian gives his 'KUET roll no' and 'batch name' the terrorist will say 'We are the Worst. Forgive us!'

'Jack is back'!

Using Matlab:

```
1)  disp ( ' Jack is missing ')
2)  if  ( jack = = 'good dog' )  &&  ( age > 2 )
        permission = 1;
    end
3)  % 2 routes = ' while loop '  and  ' for loop '
    for cave= 1 : length ( [ cave1  cave2  cave3 . . . . . caveN])
        if cave_color ==  ' red '
                disp ( ' Jack is found ! ')
                break;
        end
        if cave_color == ' blue '
                continue;
        end
        disp ( ' Jack not found. Searching next cave ')
    end
```

```
4)  testimony ( your_batch, your_roll )
    function testimony (batch, roll )
        if ( batch > 15) && (batch < 20)
                if length (roll) == 7
                        disp  (' We are the
worst. Forgive us!')
                end
        end
    end
```

Competitive Programming

Step 1

Given Problem Statement/ Case

»»

Find the inner product of two vectors by using a function name ‘ inner ’.

Step 2

Your scratch algorithm

»»

What is inner product?
If x=[1 2 3], y= [4 5 6]
Then inner product? –> dot product

Step 3

Your code/program

»»

```
function z = inner(x,y)
        z = dot (x , y);
end
```

Step 4

Checker Function

»»

```
x = 1 : 3 ;
y= 3 : -1 : 1;
z_correct = 10 ;
assert(isequal(inner(x,y), z_correct))
```

Advaned Programming

Advance works (outline):

- Congrates! You can now dive into more Sophisticated works with Matlab

Plotting curve	Algebric Equations	Differentiation
Integration	Calculus	Polynomials
Laplace and Fourier transform	Machine Learning + Neural Network	Simulink : Simulation and System Design

- Best resources: Mathworks Website and Matlab 'Help'
- Best practices: Matlab Cody Challenge

Not the best

Good for beginners

- Reference: https://www.tutorialspoint.com/matlab/index.htm

Plotting Curve

Syntax:

```
x=[0 : 0.01 : 10];                          ! X-axis data
y= sin (x)                                  ! Y= f(x)
plot (x , y, ' r- ')                        ! Plot y vs x with dash ( - ) and of red color (r)
xlabel(' your x-axis label')
ylabel('your x-axis label')
title('title of the plot')
legend (' legend of the plot' )             ! Will add string a with string b
grid on                                     ! To show grid
axis equal                                  ! Both x and y axis will have similar spacing
axis ([xmin xmax ymin ymax])                ! Show plot within range x = xmin : xmax
                                            ! And y = ymin : ymax
subplot ( r, c, n)                          ! Create subplots of r no of row, y no of
                                            ! column. And n = your plot position no
```

More on the topic:
https://www.tutorialspoint.com/matlab/matlab_plotting.htm

Plotting Curve

Graphics:

1.) Drawing bar chart :

```
bar (x , y)                    ! x = range, y = value at those x points
```

2.) Drawing contours:

```
meshgrid (starts : increment : end)
contour (x_range , y_range ,  y)
```

```
! x_range = a range of x_point
! y_range = a range of y_point
! y = f(x)
```

3.) 3D (Three dimensional plot):

```
meshgrid (starts : increment : end)
surf (x_range , y_range ,  y)
```

```
! Should be same dimension for x & y
! x_range = a range of x_point
! y_range = a range of y_point
! y = f(x)
```

More on the topic:
https://www.tutorialspoint.com/matlab/matlab_graphics.htm

Algebric/polinomial Equation

Graphics:

Two basic ways:

Using solve:

```
solve ( ' equation=0 ', var )
```

Using root:

```
roots ([ coefficient from highest to lowest power of variable])
```

Algebric Equation :

```
x^4 - 7*x^3 + 3*x^2 - 5*x + 9 = 0
```

Matlab code:

```
solve ( ' x^4 - 7*x^3 + 3*x^2 - 5*x + 9 = 0 ' , x )
roots ( [1  -7   3  -5  8 ])
```

https://www.tutorialspoint.com/matlab/matlab_algebra.htm

Calculus : Diff.+Int.+limit

Differential:

```
diff ( eqn  , respective_variable)
diff ( x^4 – 7*x^3 + 3*x^2 – 5*x + 9 = 0 , x) ! Means derivating the
                                                ! equation w.r.t  x
```

Integral:

```
diff ( eqn  , respective_variable)
int ( x^4 – 7*x^3 + 3*x^2 – 5*x + 9 = 0 , x)  ! Integration w.r.t  x
                                                ! without limit
int ( x^4 – 7*x^3 + 3*x^2 – 5*x + 9 = 0 , x , –3, 5)  ! With limit –3 to 5
limit ( eqn, var, limiting_point)                      ! Finding limit at a point
```

https://www.tutorialspoint.com/matlab/matlab_calculus.htm

Laplace and fourier transform

laplace:

$$L\{f(t)\} = \int_0^{\infty} f(t).e^{-st}\,dt$$

laplace (time_domain_eqn) ! Converts to ‘s‘ domain
ilaplace (s_domain_eqn) ! Inverse laplace coverts to ‘t‘ domain

Fourier:

fourier (time_domain_eqn) ! Converts to ‘f‘ domain
ilaplace (f_domain_eqn) ! Inverse Fourier coverts to ‘t‘ domain

- These transforms are vastly used in harmonic analyzing, image and signal processing

What's next?

Two highly demanded topics of Research works

Machine Learning + Neural Network

https://www.mathworks.com/discovery/neural-network.html

https://www.mathworks.com/discovery/neural-network.html

Simulink : Simulation and System Design

https://www.mathworks.com/products/simulink.html

Keep an eye on Makers Hub's future events on these topics

Thank you all