

%% ----- ALL ABOUT MATRIX ----- %%

```
// Creating a array
$ A = [1,2,3;2,3,4;6,7,8]
    [ ] -----> concatenation operator
    ,   -----> concatenate row wise
    ;   -----> concatenate column wise

// Finding array size
$ [m,n] = size(A)
m = size(A,1) //returns row size
n = size(A,2) //returns column size

// Finding total element of a matrix
totalElement = numel(A)
totalElement = size(A,1)*size(A,2) // also returns the total element

// Selecting last element
$ A(end)

// Selecting last column
$ A(:,end)

// Selecting specific columns
$ A(:,[2,5]) //selects 2nd and 5th column

// Upper and Lower Bounds of a matrix
$ [min,max] = bounds(A)
Min = min(A)
Max = max(A)

// Reduced Row echelon form
$ R = rref(A)

// Row Echelon Form
$ [L,U] = lu(A); //LU factorization
R = U;
```

```

%% ----- ALL ABOUT PLOTTING ----- %%

// Taking a figure and assigning a name to it
$ f = figure(1);
  f.Name = 'This is the figure name';

// How to take data points
$$ x = linspace(range,size) // range given in matrix.default size 100
$ x = linspace(-5,5,200) // creates 200 point between -5 and 5

// Basic Plot
$ plot(x,y) //x and y are two vectors of same dimension

// Plotting several matrix against one matrix
$ plot(x,mat1,x,mat2) //plots mat1 and mat2 againsts x

// Plot Properties
$ p = plot(x,y);
  p.LineStyle = "--";
  p.Marker = "o";
  p.Color = [1,0,0];

// Axis Properties
$ plot(x,y)
  axis = gca;
  axis.XTick = [10,45,70];
  axis.YTick = [.2,.8];
  axis.XLim = [0,20];
  axis.Title.String = "Hello Title";
  axis.XLabel.String = "";
  axis.YLabel.String = "";

// 3D Plotting
$ x = linspace(0,10,50);

```

```

y = linspace(0,20,60);

[X,Y] = meshgrid(x,y);

Z = cos(X)+sin(Y);

s = surf(X,Y,Z);

```

```

%% ----- ALL ABOUT MISCELLANEOUS ----- %%

// Taking 23 random real numbers between 67 to 78
$$ A = rand(size)
$ A = rand(1,23); //returns number between 0 and 1
A = A*(67-78) + 67; //shifts numbers between 67 and 78

// Taking 23 random integer numbers between 67 to 78
$$ A = randi(range,size)
$ A = randi([67,78],[2,4])

// Solving a system of equations
$$ S = solve(equations,variables)
$ syms x y
Eq1 = X + y == 100;
Eq2 = X - y == 40;
S = solve([eq1,eq2],[x,y],"Real",true);
S = solve([eq1,eq2],[x,y],"ReturnConditions",true);

// Solving system of linear equation
$ syms x1 x2 x3 x4

eq1 = 2*x1+x2+x3-x4 == 12;

eq2 = x1+5*x2-5*x3+6*x4 == 35;

eq3 = -7*x1+3*x2-7*x3-5*x4 == 7;

```

```

eq4 = x1-5*x2+2*x3+7*x4 == 21;

[A,B] = equationsToMatrix([eq1,eq2,eq3,eq4],[x1,x2,x3,x4]);

X = linsolve(A,B)

// Solving equations numerically
$ syms x y
Eq1 = x+y
Eq2 = x-y
// S = vpsolve(equations,variables,range) //
S = vpsolve([eq1,eq2],[x,y],[-10,20])

// Simplifying expressions
$ simplify(expression);

// Finding differentiation
$$ diff(function,wrtVariable,numberOfDiff)
$
Syms f(x,y)
f(x,y) = 3*x.^2+2*y;
diff(f(x,y),x,2);

// Finding integration
$$ int(function,wrtVariable,range)
$ Syms f(x,y)
f(x,y) = 3*x^2+2*y;
int(f(x,y),x,[0,1]) // integrates with respect to x

```

```

// Difference between expression and function
$ syms x f(y)

G = x^7; //expression

F(y) = y+2; //function

G(2) //Error as this is an expression

F(2) //4 as this is a function


// Single line Functions
$$ @(argumentList) expression
$ f = @(x,y) x+y;

f(2,3) //5 as it returns the value 5. Is not symbolic type.

// Calculating double integral
$$ integral2(fun,x0,x1,y0,y1)
$ fun = @(x,y) 1./ ( sqrt(x + y) .* (1 + x + y).^2 );

ymax = @(x) 1 - x;
q = integral2(fun,0,1,0,ymax)


// Solving second order ODE
$$ X = dsolve(odes)
$$ X = dsolve(ode,cond)
$ X = dsolve(ode,[y1==0,y2==2])


// Doing interpolation
$$ interp1(X,Y,queryVal,method)
$ interp1(x,y,23,'Linear') //splines,pchip → other methods


// Doing extrapolation

```

```
$$    interp1(X,Y,queryVal,method,'extrap',defaultValue) // defaultsValue is
the value of Y outside the range of X

$    interp1(x,y,63,'Linear','extrap') //splines,pchip → other methods

//    Creating a table
$    T = table(columnVector1,columnVector2.....);
      T.Properties.VariableNames = ["heading 1","heading 2"];
```