Summer Of Software

* Matlab On Ramp
* MATLAB is a programming and numeric computing platform.
* MATLAB abbreviates for Matrix Laboratory.
* Commands
* In Matlab we can directly compute basic mathematical operations such as +, -, +\*, /.

Ex. 3+5 then pressing enter directly prints ans =8.

* By default, in matlab if variable is not defined then it automatically stores the value in variable named “ans”.
* = is called assignment operator. It assigns the value on the right to the variable on the left.
* All the variables declared in the command window are stored in the workspace window in the right.
* By using up arrow key, we can also return to some command used above in command window and can also edit it.
* We can name variables anything in matlab as long as they start with a letter and contain only letters, numbers, and underscores (\_).

Ex: 3sq is invalid variable name, while sq3, s\_3q, sq3\_, sq\_3 are all valid variable names.

* We can save variables in workspace to a MATLAB specific file format called a MAT-file using the save command or function.
* We can remove all the variables from workspace with the clear function.
* To load variables from a MAT-file, use the load command.
* The clear function cleans up the workspace. we can use the clc command to clean up the Command Window.
* MATLAB also has some in-built functions ex: abs, sqrt, eig and constants like pi=π=3.1416.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

* Format command
* Vectors and Matrices
* A single number, called a *scalar*, is actually a 1-by-1 array, meaning it contains 1 row and 1 column
* You can create arrays with multiple elements using square brackets.

x = [3 5] or x= [3,5].

* When we separate numbers by spaces MATLAB combines the numbers into a row vector, which is an array with one row and multiple columns (1-by-n). but when you separate numbers by semicolons, MATLAB creates a column vector (n-by-1).

x = [1;3] x = 1

3

* we can combine spaces and semicolons to create a matrix, which is an array with multiple rows and columns. When entering a matrix, we enter them row by row.

x = [3 4 5;6 7 8]

x= 3 4 5

6 7 8

* Creating evenly spaced vectors
* For long vectors, entering individual numbers is not practical.

An alternative, shorthand method for creating evenly spaced vectors is to use the : operator and specify only the start and end values.

Ex y = 5:8

y = 5 6 7 8

Notice that square brackets are not needed when you use the colon operator

* The : operator uses a default spacing of 1, however we can specify our own spacing. For example, to use a spacing of 2 use

x = 20:2:26

x = 20 22 24 26.

* If we know the number of elements we want in a vector (instead of the spacing between each element), you can use the linspace function

i.e., linspace (first, last, number\_of\_elements).

Note the use of commas (,) to separate inputs to the linspace function.

x = linspace (0,1,5)

x = 0 0.250 0.500 0.750 1.000

* Both linspace and the : operator create row vectors. But if we need a linearly spaced column vector? The transpose operation (') converts a row vector into a column vector.

x = 1:3

x =1 2 3

x = x'

x = 1

2

3

* Random vectors and matrices
* We use functions rand, ones, zeros.

Table

Description automatically generatedText

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

* Indexing and modifying arrays
* We can extract values from an array using row, column indexing.

Ex. y = A (5,7) This syntax extracts the value in the 5th row and 7th column of A and assigns the result to the variable y.

* we can use the MATLAB keyword end as either a row or column index to reference the last element. And also use end arithmetically as (end-1, end-2)

y = A(end,2)

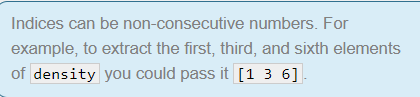
Graphical user interface, application

Description automatically generated

* When used as an index, the colon operator (:) specifies all the elements in that dimension. The code

x = A(:,1) - creates a column vector containing all of the elements from the first column of A.

* You can use the colon operator to specify a range of values. The following code creates a matrix containing the first, second, and third rows of the matrix A. x = A (1:3,:).
* A single range of index values can be used to reference multiple vector elements. Ex. x = v(3:end) - returns a subset of vector v containing the elements from 3 to the end.



* We can also use indexing to change the values in a matrix.

Ex. A(2,3)=10 changes 2nd row 3rd column element to 10.

Text

Description automatically generated

* Array Calculations
* we can add a scalar value to all the elements of an array.

x = [1 2 3]

y = x + 2

thus y = 3 4 5

* we can add together any two arrays of the same size. z = x + y
* we can multiply or divide all of the elements of an array by a scalar.

Ex. Z=2\*x

* The maximum value of a vector can be determined using the max function.

v = max(x)

* We can also use functions which work on complete vector in single command ex. Sqrt and round.
* The .\* operator performs elementwise multiplication and allows you to multiply the corresponding elements of two equally sized arrays.

z = [3 4] .\* [10 20]

z = 30 80.

* The size function can be applied to an array to produce a single output variable containing the array size in a two-element array, where the first element is the number of rows, and the second element is the number of columns. Ex. s = size(x)
* The size function can be applied to a matrix to produce either a single output variable or two output variables. Use square brackets ([ ]) to obtain more than one output. [xrow,xcol] = size(x).
* You can find the maximum value of a vector and its corresponding index value using the max function.

The first output from the max function is the maximum value of the input vector. When called with two outputs, the second output is the index value. [xMax,idx] = max(x)

Graphical user interface, text, application

Description automatically generated

* Plotting Data
* Two vectors of the same length can be plotted against each other using the plot function. Ex. plot(x,y)
* we can use another input to the plot function to specify the colour, line style, and marker style using different symbols in double quotes.

Ex. plot(x,y,"r--o") This command plots a red (r) dashed (--) line with circle (o) markers.

* To plot one line on top of another, use the hold on command to hold the previous plot while you add another line.
* While the hold state is on, plots will continue to go on the same axes. To return to the default plot behaviour, where each plot gets its own axes, enter hold off.
* When we plot a single vector by itself, MATLAB uses the vector values as the y-axis data and sets the x-axis data to range from 1 to n (the number of elements in the vector). Ex. plot(y)
* The plot function accepts optional additional inputs consisting of a property name and an associated value. Ex. plot(y,"LineWidth",5)

The command above plots a heavy line.

* Labels can be added to plots using plot annotation functions, such as title.

Ex. Title(“xyz”)

xlabel(“x axis label”)

ylabel(“y axis label”)

legend(“abc”)

* Importing data
* To extract a variable from the table we can use dot notation.

data.VariableName

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* Logical Arrays
* Relational operators, such as >, <, ==, and ~= perform comparisons between two values. The outcome of a comparison for equality or inequality is either 1 (true) or 0 (false).
* we can use a logical array as an array index, in which case MATLAB extracts the array elements where the index is 1 (true). The following example will extract all elements in v1 that are greater than six.

v = v1(v1 > 6).

* we can use logical indexing to reassign values in an array. For example, if we want to replace all values in the array x that are equal to 999 with the value 1, use the following syntax. x(x==999) = 1.
* Graphical user interface

  Description automatically generated with low confidenceProgrammingGraphical user interface, text

  Description automatically generated

Graphical user interface, application

Description automatically generated