Lec03 IAM550 J.Raeder 9/3/2019 MATLAB Programming

The following list is MATLAB 'stuff' you should learn in order to get started with MATLAB. We will go through much of this in class, and you can find more/similar detail on the web. It will be well worth your while to spend some time with MATLAB open, going through the examples as they are described.

- MATLAB Command window
    - The primary place you interact with MATLAB.
    - Prompt: >> (MATLAB is ready and waiting)
- Scalar (single number) variable assignments
    - Defining your own variable names
        - No spaces! Lead with a letter
        - Case sensitive
        - Numbers ok (after the first character)
        - Punctuation is not ok except underscore: a_123_variable
        - Is there a limit on the variable size? Yes, but very large
    - '=' is the assignment operator
        - It is also the 'replacement operator – be careful of clobbering yourself.
        - Note that you can only have one variable on the left hand side
            - Try typing x+2=20 in the MATLAB command window
            - x = x + 2 +sin(x) is OK
    - MATLAB workspace
        - MATLAB commands 'who' and 'whos'
        - The 'exist' command
    - MATLAB assigns everything to a variable
        - Default: ans.
    - Predefined constants (note that you can redefine these, although that's generally not a good idea!)
        - ans: default variable name
        - pi: ratio of the circumference of a circle to its diameter)
        - eps: smallest number such that when added to one creates a number bigger than one according to the computer
        - i,j: square root of -1
        - inf: 1/0
        - NaN: Not-a-Number – 0/0
        - realmin: the smallest usable positive real number
        - realmax: the largest usable positive real number

- Command window
    - The ‘clear’ command → clears window
    - The ‘clc’ command → removes all variables
    - Help (topic)
    - Quit → exits MATLAB
    - Precision (how numbers are printed)
        - Format short
        - Format long
        - Format bank
        - Format long e
        - Format short e
        - (format +, format rat)
- cd, chdir *dir* → change current working directory
- mkdir *dir* → make new directory
- pwd → show current working directory (explain absolute vs relative path)
- diary *file* → opens diary file
- edit *file* → open file editor (later for scripts)
- ls –al → long file listing in cwd

more commands on https://www.tutorialspoint.com/matlab/matlab_commands.htm

- Arithmetic operations
    - Exponent, addition, subtraction, multiplication, division
    - Order of precedence
        - Expressions are evaluated from left to right
        - Highest to lowest precedence
            - power operation (2^3*2 vs. 2^6)
            - multiplication and division
            - addition and subtraction
                - 3*2+1 vs 1+3*2
        - Parentheses can be used to alter precedence
            - Innermost, proceeding outwards
            - If you are unsure about precedence, parentheses don’t hurt you (and they probably make your code more readable)
- Punctuation
    - Semicolons suppress screen printing, and can be used to separate commands. (they also denote a new row in an array)
    - Commas can be used to separate commands, and also separates elements of an array

    - … (ellipsis) continues a line (particularly useful when writing long expressions in scripts)
- Comments
    - Helps people (including you) understand your scripts
    - Any text preceded by a percent sign (‘%’) is ignored by the MATLAB interpreter
- Arrays (intro, only 1d)
    - MATLAB is really good at handling large collections of numbers (arrays) as if they were a single variable
    - Arrays: collection of numbers in some meaningful (to you, the programmer) collection of numbers. More precise: indexed set of numbers (or strings, later).
    - To define an array, surround the numbers by brackets and separate using spaces or commas → array1 = [1, 5, 7, 22, 31]; array2=[4 7 9 22.3]
    - A=linspace(22,33,100) → 100 elements evenly spaced from 22 to 33
    - B=logspace(44,55,71) → ditto with log spacing
    - The ‘length’ function tells you how many values are in an array → l1=length(array1)
    - Appending vectors: a=[1,2,3]; b=[3,4,5]; c=[a,b]
    - You can address a particular value using a parenthesis with an *index*
        - E.g., x = [1, 3, 7, 2, 11]; x(3) → 7
        - In MATLAB the first index is always 1 (C, Perl, Python: 0; Fortran → also 1, but can be redefined)
    - The colon command and regularly spaced numbers in an array → A = 1:8 same as A = [1,2,3,4,5,6,7,8]; likewise B=1:3:11 → same as B=[1,4,7,10]
    - Scalar-array addition and subtraction can be done using + and – → C=B+5
    - Also scalar multiplication/division: D=5*A; E=B/7
    - Element-by-element multiplication and division can be done using .* and ./ operators → A=[1,2,3]; B=[4,5,6]; C=A.*B; D=A./B
    - Vector dot product → c=dot(a,b)
    - Vector magnitude → a[1,2,3]; b=a.*a;m=sum(b); mag=sqrt(m)
    - Or → mag=sqrt(dot(a,a))
    - Or → mag=sqrt(sum(a.*a))
- Built in functions – there are lots! For example:
    - sin and sind, cos and cosd (rad/degrees, avoid the degree versions)
    - exp
    - sqrt
    - log and log10
    - acos, asin, atan
    - all built-in functions also work on arrays → a=1:0.1:10; b=sin(a)

    - Note that you can (and will) make your own later on functions in the course
- Plot command: plot(x,y)
    - Opens a separate graphics window
    - x,y are arrays of same lenth, plots y as a function of x

```
x = [0:5:100];
y = sin(x);
plot(x, y)
```

    - Some useful commands to go along with plot:
        - xlabel
        - ylabel
        - title
        - grid

```
x = [0:0.01:10];
y = sin(x);
plot(x, y), xlabel('x'), ylabel('Sin(x)'), title('Sin(x) Graph'), grid on, axis equal
```

        - text string must be in single quotes
    - changing how your graph looks:
        - Either change this in the command…
            - plot(x,y,’:‘)
        - …or change the properties in the figure
    - Plotting two lines in the same graph:
        - plot(x1,y1,x2,y2)
        - Or, use the ‘hold’ command

More at: https://www.tutorialspoint.com/matlab/matlab_plotting.htm

- MATLAB help
    - Typing ‘help functionname’ gives you syntax, and explanation, and often an example. It also should give you a link to the help browser.
    - The help browser can help you navigate if you are unsure of the specific function you are looking for
        - It also contains tutorials, which are well worth looking at
    - Check out www.mathworks.com/matlabcentral/
        - User file exchange
        - MATLAB answers