An Introduction to Programming through MATLAB

1. Steps in Programming or Programming in Steps! [1]

    a. Decompose a Problem – break a large problem into smaller problems

    b. Specify the problem, develop an algorithm in terms of the most basic steps, write the pseudocode, type the program, run it and analyze results

    c. Make sure that algorithm is well-defined, unambiguous, effective, and finite. Algorithms can be optimized for speed, efficiency, readability, typing time, length, etc.

2. MATLAB layout

    a. Current Folder Window – this lists the files in the current folder

    b. Details Window – this lists the details of the selected file listed in the current folder

    c. Command Window – this window allows you to enter commands and observe the output. To clear the contents of Command Window use the clc command. To suppress output in the command window, end your statement with a semi colon.

    d. Workspace Window – this window summarizes all your current variables. You can choose the items you would like to see in the summary by right clicking on the top bar of this window and checking the values you would like to see. There is also a quick access menu in this window to create, open/edit, delete, and import variables. To clear a variable use the clear command. To clear all variables, use the clear all command.

    e. Variable Editor Window - Double clicking on a variable opens the variable editor window where you could observe the details of the variable and edit it by force. You can also copy and paste Excel data in the variable editor window.

    f. Command History Window – this contains the list of commands entered in the command window and is retained over sessions.

    g. To create a new MATLAB program, click on 'New Script' or enter <CTRL> <N>. A MATLAB program is a set of commands that you could potentially have also entered in the Command History Window.

3. Variables, constants, and arrays

    a. A variable is just like a variable in middle school algebra. At any given moment, a variable has a fixed value but you can change the value in the program – hence the name 'variable'.

    Rules for variable names in MATLAB: A valid variable name starts with a letter, followed by letters, digits, or underscores. MATLAB® is case sensitive, so 'A' and 'a' are not the same variable. The maximum length of a variable name is 63 characters[2]. You cannot define variables with the same names as

[1] In this class we will focus on Procedure Oriented Programming rather than Object Oriented Programming because though the latter is often more useful for certain types of agent-based modeling, the former is more broadly applicable for the types of models we will solve.

[2] This may be different for different versions. To find the maximum length for your version, run the namelengthmax command.

MATLAB keywords, such as if or end[3]. Avoid creating variables with the same name as a function (such as size, sum, or max). In general, variable names take precedence over function names. If you create a variable that uses the name of a function, you sometimes get unexpected results.[4]

Variables can be of different types – integers, floating points (decimals), logical, etc.

b. A constant is typically a variable whose value you do not change within the program. MATLAB provides some helpful 'constants' such as realmin, realmax, eps, inf, pi,etc.

c. An array is like a subscripted variable that allows for the same variable to hold a different value for each subscript. An array can be one dimensional, two dimensional, or multi-dimensional. A one dimensional array is also known as a vector. A two dimensional array is a matrix. MATLAB allows us to distinguish between rows and columns. MATLAB allows a great deal of functionality for various matrix operations. Often, MATLAB assumes matrices to be two dimensional by default.

    i. Matrices can be created through square brackets, commas or spaces (to separate columns), and semicolons to separate rows. For instance, X = [3,4,5; 2 6 7] creates a 2X3 matrix.

    ii. Matrices can also be created with zeros, ones, rand, eye, magic, etc. For instance, Y = rand(3,4) creates a 3X4 matrix with uniformly distributed random numbers between 0 and 1.

    iii. To refer to a particular element in an array, use the array name immediately followed by parenthesis with the subscripts. For instance Y(1,2) refers to the element in Y that is in the first row and second column.

    iv. MATLAB uses colons with great effect. : by itself indicates from first to last. 1:10 indicates start at 1 and end at 10 incrementing by 1. 1:2:10 indicates start at 1, increment by 2, and end at 10. Colons can be used to create row vectors or to make references to subsets of elements. Z=4:-1:2 creates a row vector Z = [4 3 2]. Z(2:3) will return the elements in the subset of Z starting from 2 and ending at 3. So Z(2:3) will return the elements 2 and 3.

4. Arithmetic operations

MATLAB performs all the standard arithmetic operations +, -, *, /, ^. This also works for matrices when applicable. If you use 'X*Y' MATLAB will provide the cross product of matrix A and B – note that the order matters! If you use dot(X,Y) MATLAB will provide the dot product of X and Y. If you use '.*' MATLAB will provide element by element multiplication. You can also do element by element multiplication of two or more arrays of the same size with '.*'. Please note that if you multiply or divide an array with a scalar then by default MATLAB does element by element operations. 3*X will return all the elements of X multiplied by 3. MATLAB can also do matrix division (i.e., it multiplies by the inverse). For a complete list of operations, refer to https://www.mathworks.com/help/matlab/arithmetic-operators.html

The values of variables can be assigned through arithmetic operations. For instance X= X*Z.

[3] For a complete list of keywords, run the iskeyword command

[4] Check whether a proposed name is already in use with the exist function. For instance, type exist sum to see if the name sum is currently a function or existing variable.

5. Conditional statements

   Unlike arithmetic operations, conditional statements answer questions. To create a conditional statement you need to use >, <, <=, >=, or ==.

   In MATLAB only the value of 0 is considered false. While any other value is considered true, the default value for true is 1.

   2>3 returns a value of 0. 2==3 returns a value of 0. 2<=3 returns a value of 1. 2<3 returns a value of 1.

6. Boolean or logical operations

   MATLAB performs all the standard logical operations & (for vector and), && (for scalar and), | (for vector or), || (for scalar or), ~ (for not).

   1 && 1 will equal 1; 1 && 0 will equal 0; etc.

   Similarly, x>y && a>b will return 1 only if x is less than y and a is greater than b.

   For vectors, you can also use all(X) which returns 1 only if every element in X is non-zero. Similarly you can use any(X) which returns 1 if any element in X is non-zero.

7. Control flow and loops

   a. if, elseif, else, end – use this set of commands to go through optional blocks of code

      For example the following block of code that returns the sign of the variable X in the variable A[5]:

```
if X>0
        A= 1;
elseif  X<0
        A=-1;
else
        A=0;
end
```

   b. for end – use this to repeatedly execute a set of commands for a <u>fixed</u> number of iterations.

      For example the following block of code sums up the integers from 1 to a 100.:

```
s=0;
for i=1:1:100
        s=s+i;
end
```

[5] Please note that while writing scripts, MATLAB automatically indents. If you mess up the indentation, you can also indent automatically by pressing <CTRL> <I>

Please note that MATLAB allows us to use statements such as:

```
s=0;
for i=X
        s=s+i;
end
```

In this case, i will cycle through all the elements of the array X. s will return the sum of the elements of X.

A particularly useful for loop is the following:

```
for i=randperm(52)
        …
end
```

In this case, i will cycle through the values from 1 to 52 in random order.

c. while end - use this to repeatedly execute a set of commands until some condition is satisfied.

   For example the following block of code computes the factorial of n and stores it in F.

```
F=1;
n=5;
while n>1
        F=F*n;
        n=n-1;
end
```

d. break can be used to break out of a for or while loop. Usually this is used if some condition is encountered that makes continuing the loop pointless. continue can be used to skip the intermediate steps in a loop and move on to the next iteration of the loop.

8. MATLAB functions

   a. In-built functions – MATLAB has thousands of in-built functions. Some that you will find useful include sum, mean, log, sort, std, min, fminsearch, spy, imagesc, plot, etc. To obtain help on any of these functions simply type in help followed by the function name in the command window.[6]

   b. User-defined functions – MATLAB also lets you create your own functions. We will see many examples of user –defined functions in class. There are some important naming and syntax requirements for user defined functions.

[6] The creators of MATLAB have a sense of humor. Type 'why' or 'spy' or 'image' etc. in the command window to see some evidence.

9. Miscellaneous suggestions

   a. Comment/document extensively for your own sake and for the sake of other readers of your program. To create comment in MATLAB, simply start the line with a % sign. To comment or uncomment chunks of code you can also use keyboard shortcuts <CTRL><[> and <CTRL> <]>.

   b. Version Control – remember that script or .m files are simple text files and very small in size and take up negligible room on the hard drive. Please save new versions each time you make substantial changes. You can make an 'Archive' folder to store older versions of your program. This is helpful if some change you made ruins the program and you have difficulty going back to a previous version. A MATLAB program whose latest version has not yet been saved will show up with an asterisk in its title header.

   c. Hitting <CTRL> <C> or <CTRL> <BREAK> will help force an end a program that is running endlessly (or seemingly endlessly). You can also use the Pause button in the editor window and debug tools.

   d. You can save the values of variables in .mat files. You can load these values later. You can also copy and paste data to and from Excel.