MATLAB notes

A ***numerical method*** is a complete set of unambiguous elementary mathematical operations, usually iterative, which result in the approximate solution of a problem. Knowledge of numerical methods is absolutely necessary to interpret the output of various software packages with confidence, wisdom and insight.

A computer program (source code) is a list of instructions which the computer executes. The order in which the instructions are executed is determined by logical tests within the program based on input data. Typically, certain sets of instructions are executed multiple times (loops) until a particular outcome is obtained. A program usually consists of the following parts:

- data input, either from the keyboard of from an existing data file, frequently both.
- assignment of variables within the program
- execution of mathematical operations in an order determined by the program logic
- output of numerical data and graphs.

Mathematical operators in order of priority:

Operation	Symbol	Example
exponentiation	^	3^2 = 9
multiplication and division	* /	8 * 3 / 2 = 12
addition and subtraction	+ -	8 + 3 – 2 = 9

With equal priority, operations are carried out left to right. Parentheses can be used to establish order of operations. Rules of priority are applied within each set of parentheses, starting with the innermost set of parentheses and working outward.

Variable names: Variable names are case sensitive and may contain up to 63 characters. Variable names must start with a letter, may contain numbers, and cannot contain embedded blanks. The only special character allowed is the underscore. No punctuation is allowed.

Reserved words: Certain words cannot be used as variable names since they are part of the syntax of MATLAB. Some of these words are as follows: for, end, if, while, function, return, elseif, case, otherwise, switch continue, else, try, catch, global, persistent, break.

MATLAB contains many built-in functions (e.g. sin, cos, log, etc.) whose names should not be used as user-defined function names.

Compiled vs. Interpreted Code: Computer languages generally are characterized as being compiled (e.g., C, C++, Cobol, FORTRAN, Pascal) or interpreted (e.g., Java, JavaScript, Perl, Python, C#). With a compiled language, the code written by the programmer is converted as a whole into machine language, and linked with libraries of pre-written functions (already in machine language). The final product is a stand-alone executable file (.exe) specific to the hardware and operating system under which it will be run. Advantages of compiled code are as follows: It runs much faster, the program which created it doesn't have to be installed on the machine running it, and the code is encrypted, i.e., hidden from the people using it.

With an interpreter, each line of code is converted to machine language on the fly as the program is run. This takes much longer than running pre-compiled code. An advantage is that the same code can run unmodified on different machines with different operating systems. Generally, code development and debugging is faster with an interpreter since the compiler doesn't have to be run each time a change is made.

Native Matlab is run as an interpreter, although it can be compiled. It is never completely stand-alone, however.