Anthony D’Alessandro

Applied Probability and Statistics - Introduction to MATLAB

MATLAB (Matrix Laboratory) is a high-level programming language developed by MathWorks that allows programmers to work with entire matrices and arrays. At its core, MATLAB is essentially a powerful tool for computational mathematics as it specializes in concepts such as linear algebra, non-linear functions, statistics, calculus, curve fitting, data visualization, and many others. There are many similarities between MATLAB and various object oriented programming languages such as Java including basic ideas such as variables, data types, operators, loops, and conditional statements. One detail regarding basic functionalities deserves to be noted which is the lack of need to declare data types for variables. This is similar to Python’s dynamic typing in which variable properties are determined at run time rather than compile time. Considering all of these characteristics of MATLAB, learning the basic functionalities is much easier if the user has some background in other traditional programming languages.

Some of the more advanced powerful features of MATLAB include plotting, graphics, and polynomials. Plotting is especially useful in which the *plot* command can be used with a range of values and a function to create a simple graph (e.g. plot(x, y) where x = [0:1:10] (0 - 10 with an increment of 1) and y = x (linear function)). Graphics are another practical tool to visualize data which include things such as bar charts and three-dimensional plots. The *bar* command is similar to *plot* in which it uses a range of x values but instead of a function y, it uses a list of data points that correspond with the x values (e.g. bar(x, y) where x = [1:5] and y = [10, 5, 20, 15, 10]). Another graphic tool is the *meshgrid* command that is commonly used in conjunction with the *surf* command to plot the surface of a graph in three dimensions. Considering the fact that a color-coded three dimensional graph can be created using about four lines of MATLAB code really demonstrates how powerful this language is while performing this task in other languages would take many more lines to even come close to the same result.

Another useful MATLAB feature is the ability to directly work with polynomials using the *polyval* function. While declaring a polynomial function is not something that is unique to MATLAB, the implementation is what sets it apart from other languages. For example, in Java, a variable can be assigned the value of a polynomial function such as (int y = x\*x + x + 1) where the function can be evaluated for any value of x. Using MATLAB, the same polynomial can be declared as (p = [1 1 1]) and the *polyval* function can be used to evaluate the expression at some value x using (polyval(p, x). This function has the potential to be especially useful when polynomials get increasingly more complex. MATLAB includes other variations of this function as well such as *polyvalm*, *poly*, and *polyfit* which evaluate a matrix polynomial, return the original polynomial given its roots, and apply polynomial curve fitting, respectively.

There are countless other features in MATLAB that go into more depth and cover more complex topics such as transformations, derivatives, integrals, etc. While MATLAB can be used to assist programmers in relatively complex graphing tasks and computational mathematics problems, it is commonly used in extremely intensive environments at companies and organizations such as NASA, Tesla, AMD, and various other industry leaders. With this in mind, having some fundamental knowledge of programming in general with a concentration in MATLAB is very valuable in the current job climate as the demand for engineers and developers has been steadily rising in the past few decades.

Sources

<https://www.tutorialspoint.com/matlab/index.htm>

<https://www.mathworks.com/help/matlab/getting-started-with-matlab.html>