Report on MATLAB version

**Introduction:**

This report describes the purpose of the MATLAB project, the process of learning MATLAB and the results of creating and generating three equations into graphs. Instead of making nine separate graphs (one for plotting, one for salting and one for smoothing), all the formulas had their own graph with all three actions done denoted by three colors. It will detail the graphs and their properties as well as talk about the process of creating these graphs. However, for in detail MATLAB code explanation, that can be found in the documentation folder under “MATLAB Plotter Salter Smoother Documentation”.

The report will go over all the steps taken to get to this point. Because MATLAB was new knowledge, it took a little bit of time to understand it and produce what was produced. At first, nine graphs were going to be produced, but that sounded tedious when all the data for each equation could be condensed into one. That seemed a bit of a hassle to do in Java, so it was done in MATLAB. In the process of creating the graphs, a personal MATLAB tutorial was created which can be found in the project folder out in the open named “MATLAB tutorial”. That was created off a MATLAB website and worked through personally, step-by-step and developed into a mini secondhand tutorial. This report will document that process and the results of it. The tutorial portion will be written in first-person, and the rest of the report is going to be written in third person for professional purposes. The purpose of this report is truly to show the ease MATLAB brings to statistical activities vs. Java and the Java library.

**The tutorial process (see: MATLAB tutorial doc)**

* While reading and understanding the tutorial, a personal tutorial was made step-by-step to prove understanding. (MATLAB Tutorial found in first directory). The principle you do not know it until you do it was put forth here. It shows personal screenshots and an explanation of each part to prove the tutorial was worked through and understood.
* The tutorials that helped were located at: <https://www.tutorialspoint.com/matlab/index.htm> <https://www.mathworks.com/help/matlab/getting-started-with-matlab.html>.
* While the tutorial was worked through, there was a realization. The syntax was much simpler, but the properties of MATLAB were very similar to other programming languages. For example, there were variables and for – loops. There were also math functions which Python and Java also have.
* The tutorial process started by learning how to set up either MATLAB or Octave. Octave being the very similar free version. That version was chosen. It was a series of simple installation steps like any other program.
* The next part of the tutorial explained how to use the program. It went over the workspace, the command window, the file explorer and even how to create files. Understanding these basic principles made navigating much easier.
* After that, simple mathematical operations were gone over. For example, multiplication, division, addition and subtraction. This also included some fancy values like pi and square rooting.
* Some housekeeping tasks were gone over like adding comments as well as saving, clearing and loading within the command window.
* A big portion of the tutorial for someone who never learned a programming language before was the declaring variables section. (ex: x = 50). For me, it was a piece of cake—since I’ve worked in object-oriented programming languages like Java, but for a newcomer this would be something new. It explained how to declare variables and perform operations with them.
* The next couple of areas were commands, data types, creating vectors and matrices (always good to know) and other important tools a user would need to understand how to work the program to make graphs and perform math.
* After that, it dived into the more programming side of things. That was if-statements and for-loops! Personally, I did not see the need to use them for my graph output, but with a larger process of mathematical steps, it would come in handy.
* Lastly, plotting was briefly mentioned. This was new. In the Java version of this project, it was done by exporting an .csv file from Java into excel to generate graphs. The Java library part, however, was able to create a graph like MATLAB! (more steps).
* Three equations were plotted. A quadratic, a cubic and the absolute value formula. They were put onto three graphs below in MATLAB. They were combined for efficiency and the preciseness of it.

The equation is

A graph with a line graph

Description automatically generated with medium confidence

The original function shows a quadratic. Then, the noise (or salt) is extremely visible in this one. That was done by creating a ‘salt magnitude which was 2000 here. That was added to the original times a random value and subtracted by 0.5. The movmean function was used to smooth the function back to looking pretty. It made it much more like the original. Three colors were used on purpose here to show the differences between all three functions. One the original, one the salted and one the smoothed.

The equation is

A graph with a line and a square

Description automatically generated with medium confidence

Here’s a cubic function. The original blue line is clear that it is a straight line with no noise to it. It just does its thing, completing its curve. The red line, however, is the salted form of it. It shows noise in its form as it jumps up and down. Since the function was barely visible by doing what was done to the function above, 20% of it was taken to salt it. After that, the same method of salting was done from above by adding the original by the salted version and multiplying that by a random value and subtracting by 0.5.

**Function syntax overview:**

*Note: A full view of the entire syntax can be viewed in the MATLAB documentation document or viewed directly without comments from the MATLAB files.*

* Movmean() – this function takes a moving average of the salted values over the window size. It produces the ‘smoothed’ graph.
* Max() – finds the largest value or index. It works alongside salting here, allowing a visible salted function to be generated by taking 20% of it.
* Plot()– this plots a graph. Parameters can be added like the values (x and y), the line color, the line width and other attributes.
* Legend() – this names the graph title.
* Xlabel() and ylabel() – these make it possible to name the x and y axis.
* windowSize() – this creates the window size.
* Abs() – computes the absolute value of the input.