**Plotting, Salting, Smoothing 2 (Matlab/Octave)**

Tutorials used: <https://docs.octave.org/interpreter/index.html#SEC_Contents>

<https://www.youtube.com/watch?v=TqwSlEsbObg>

**Variable definition:**

A white background with black text and numbers

Description automatically generatedDefining a variable in Octave is very simple, you can just enter the name of the variable, followed by equals, and then a value.

For example, entering “x = 5” in the command line will define a variable x with a value of 5, and will then display the variable with its value in the command window.

A black and white image of a math equation

Description automatically generatedAlternatively, entering “x = 5;” will create the variable without displaying it

Any variables that you create will be stored in the workspace, and can be called at any time by entering the variable name in the command line.

A number and symbols on a white background

Description automatically generatedA screenshot of a computer

Description automatically generated

**Vectors:**

A white background with black text

Description automatically generatedAn array in Octave is a matrix, and can be one-dimensional or two-dimensional. A one-dimensional array is called a vector, and can either be a row vector (horizontal) or a column vector (vertical).

A number and equation

Description automatically generated with medium confidenceTo define a vector, you would input multiple values inside of brackets separated by spaces for a row vector, or by a semicolon and a space for a column vector.

A number on a white background

Description automatically generatedAnother way to create a vector is with the semicolon operator. This allows you to create a vector of values between two bounds.

A number and numbers on a white background

Description automatically generatedAlternatively, adding a third value in between the two bounds will define the step size.

You can also perform mathematical operations using vectors. For example, when defining a function’s x and y values, you can initialize a vector for the x values, then create a vector for y values by using a relation.

A number on a white background

Description automatically generated

The period before the multiplication indicates that the operation is being performed on each individual element in the xval vector, as opposed to trying to multiply 2 times the whole vector.

**Matrices:**

**A number on a white background

Description automatically generated**A matrix is a two-dimensional array, and can be defined similarly to a vector. For a matrix, you would enter a number of values separated by spaces to define a row, and then a semicolon to separate the columns.

A number and equal sign

Description automatically generated with medium confidenceYou can access specific elements of a vector or matrix by calling variable name, with the position of the element in parenthesis (note that indexes in Octave begin at 1 instead of 0).

A black text on a white background

Description automatically generatedA white background with black text

Description automatically generatedA white background with black text

Description automatically generatedOctave also has many built in functions for defining matrices, such as eye(n) (which creates an n by n identity matrix), ones(n, m) (which creates an n by m matrix with every value being 1), zeros(n, m) which creates an n by m matrix with every value being 0), and rand(n, m) (which creates an n by m matrix with each value being a random number between 0 and 1).

A white background with black text

Description automatically generated

**Plotting:**

**A number and symbols on a white background

Description automatically generated**To plot points on a graph in Octave, you can call the plot(x, y) function. The variables x and y can be either single values or vectors. For example, calling plot(x, y) outputs this result:

A screenshot of a computer

Description automatically generatedOctave also has built in functions for changing the style of the graph:

* title(“string”) to add a title
* set(gca, “fontsize”, int) to change the fontsize
* xlabel(“string”) to add a label on the x-axis
* ylabel(“string”) to add a label on the y-axis
* axis([lower x bound, upper x bound, lower y bound, upper y bound]) to change the bounds of the x and y axes

A close up of a text

Description automatically generatedA screenshot of a computer

Description automatically generated

A table with text and symbols

Description automatically generated with medium confidence 36:24 in YouTube tutorial:

You can also add a third parameter to the plot function to change the style of the line, the shape of each point, and the color of each point, with each style being indicated by a single character of a string. Addtionally, you can change the size of each point by adding “Markersize” and an int value as further parameters.

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

(Calling the plot function resets any formatting made to the graph, so I had to call the other functions again)

And finally, here is the plot of the function that I used for the Java implementation of plotting:

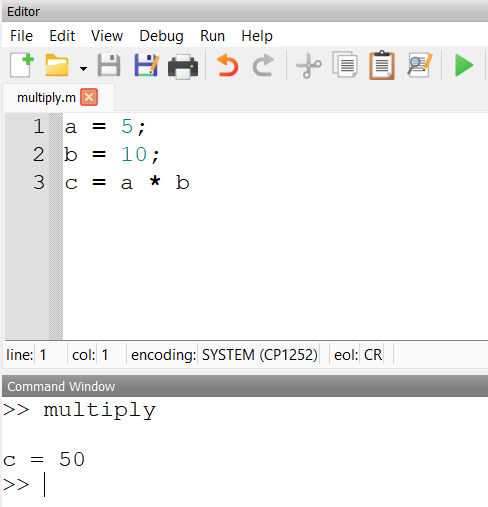
A black text on a white background

Description automatically generatedA screen shot of a graph

Description automatically generated

**Scripts and Functions:**

Scripts and funcitons are collections of commands that are stored as text file with the extention .m, and can be called from the command line to run their commands. The difference between them is that scripts have no input (their output is pre-defined) while functions require an input and return an output based on the input.



To create a script, you just type out commands line by line in the script/function editor that is built in to Octave.

For example, this is a simple script for multiplying 5 times 10. Every time the script is called, the result will always be 50.

A screenshot of a computer

Description automatically generated

Functions are created in the same was as scripts, but the file must begin with “function [output] = functionName(input)” (where output and input are both one or more variables) and end with “end” as the last line.

For example, this a function that multiplies two variables together.

A screenshot of a computer program

Description automatically generated

**Conditional Statements:**

Octave also has conditional (if/elseif/else) statements that you can use in scripts and functions.

Like functions, they must end with “end”, however if you have an if/else statement inside of a function, you can also use “endif” to specifiy that it is the if statement being ended, not the entire function

Here is an example of a function that uses an if/else statement to print if a number is positive, negative, or zero.

A screenshot of a computer program

Description automatically generated**For loops:**

Octave also has for loops that you can use in scripts and functions. For loops in Octave work by iterating through each index of a row vector, and performing some set of commands for each index.

For example, the for loop in this script iterates for each index of the row vector “i”, which is defined as a row vector consisiting of values from 1 to 4.

Thus, it multiplies the values of vectors x and y at each index i together, and adds them to index i of a new vector z.

**Plotting, Salting, Smoothing:**

A screen shot of a graph

Description automatically generatedA screenshot of a graph

Description automatically generatedA screen shot of a graph

Description automatically generatedFinally, here are the outputs of the functions that I wrote for the Octave implementation of plotting, salting, and smoothing: