Kyle Antczak

Octave Tutorial

To learn the basics of Octave I used this YouTube video: <https://www.youtube.com/watch?v=TqwSlEsbObg>

The video was very informative and covered a multitude of topics in its hour and a half runtime. This is what I learned about octave from that video.

Some useful command line functions that I thought were notable:

clc – which could be used to clear the command prompt

ans – which can refer to the previous operation output

help [function] – which tells what a built-in function does

How to preform basic arithmetic operations of addition, subtraction, multiplication, exponents, and division. Performing them as

a+b, a-b, a\*b, a^b or a\*\*b, a/b.

Declaring variables by

variable name = value

Variable names cannot start with numbers and cannot contain special characters like !. Variables are also case sensitive so variable abc id different from variable Abc. For value unlike java, text is done by using single quotes ‘value’ instead of double quotation. Also, unlike java you do not need to define the data type.

Creating vectors and some vector operations. Row vectors by

variable name = [element1 element2 ..]

with only a space between values and column vectors by

variable name = [element1; element2; ..]

with a semi colon between values. To call an element in a vector you do

variable name(j)

where j is the index. This is different from java because indexing starts at 1. There are also some other ways to create vectors with out listing out all the point like

variable name = j:i:k

where j is the starting value, k is the ending value and I is the increment between values. If you don’t include the I value so its just j:k it automatically increments by 1. One interesting way of creating a vector which is useful in the plotting points program is

variable name = linspace(j, k, m)

where j is the starting value, k is the ending value and m is the number of columns. This is useful in plotting by being able to customize number of points wanted for graphing.

Creating matrices are very similar by

Variable name = [ele1 ele2 ele3; ele4 ele5 ele6]

where values in a row are separated by spaces and different rows are separated by a semi colon.

Variable name(n,m)

Calls an element of a matrix.

Some interesting functions used to create matrices are:

eye(n) – creates an identity matrix

ones(n) – creates an all one matrix

zeros(n) – creates an all zero matrix

rand(n) – creates a matrix filled with random numbers between 0-1

(where n creates an n x n matrix)

One use full method I saw was reshape(x,n,m) which is used to rearrange the elements of x into n X m matrix.

Matrix operation of addition, subtraction, scalar, and multiplication can be done by

A+B,A-B,#\*A,A\*B

But this only works when following matrix operation rules like addition and subtraction matrices need to be the same size and for multiplication rows of A = columns of B.

Another interesting matrix operation is element by element operations for example squaring each element in a matrix. This cannot be done by normal scalar or multiplication operations and needs to be done by

A.^2

Element by element multiplication and division of 2 matrices can also be done by

A. \*B, A. /B

Plotting can be done with the operation

plot(x,y)

If x and y are numbers, it will plot the single point. You can also plot multiple points by using that function with x and y as vectors. There are a lot of methods that can change the display of the graph, some that I think are important are:

grid on/off – which is used to turn the graphs grid on or off

title(‘plot x vs y’) – which gives the graph a title

set(gca,’fontsize’,24) – which changes the fontsize

xlabel(‘variable x’) – which sets a label to the x axis, ylabel() does this for y

axis(xmin xmax ymin ymax) – is used to change the size of the axis

Using scripts and functions. Scripts take no input while functions do. To make one use the editor tab and save the file. To run them put the file name into the command prompt. To add comments similar to java instead of // any thing after % is a comment. To do a function its

function output = function name(input1,input2…)

commands

end

where output is what is returned and inputs are the input values.

User input can be done by the method

Input variable name = input(‘Enter value here:’);

If statements can be done by

If condition

commands

else if

commands

else

commands

end

and for loops as

for index = [row vector of i]

commands

end

By learning all this and playing around in octave with these concepts I feel confident in my understanding and abilities to code in octave.