APM 153 LECTURE THIRTEEN – What is and is NOT an Algorithm, Adding Information to Plots, Using the Ones, Zeros, and Eye Functions, Editing Matrices Using Subscripts, MAT files.

What is and is NOT an Algorithm

(1) From the first week of class the three beginning steps to algorithm development are,..

A. define the problem
B. determine the inputs and outputs
C. write the algorithm

(2) A and B above are necessary steps to get started, but **they are not** the algorithm. Steps A and B only define what the algorithm is supposed to do and what data it will use and what data it will produce.

(3) Steps A and B **say nothing** about **how** we will accomplish the required task.

(4) The **algorithm** is the list of the steps **that describes how** we will accomplish a task.

(5) The algorithm **is independent** of any programming language we might use to implement the algorithm.

(6) Algorithms therefore, **do not include** documentation, definitions, help files or any other stuff we might add to a program. **"Documentation" is not part of an algorithm**.

(7) Most algorithms we might write are going to **begin with input**.

(8) In the case of an algorithm we will write as a **script file**, the first lines of the algorithm will therefore be,…

Prompt the user for the input
Input the variables

(9) If we are writing a function, the first line of the algorithm will be,…

Call the function with the following input arguments,….

(10) You can prompt the user for additional input in a function, but usually, the function gets all of the data it will use as input arguments.

(11) However, in a function, often you must use the data in the input arguments to **assign values** to some of the variables used in the function. For example in root3.m we had the following lines of code.

```
function root3(coef, xn)

a = coef(1)
b = coef(2)
c = coef(3)
d = coef(4)
```

(12) Therefore, the first two lines of the algorithm for root3.m should be,…

Call function with vector of coefficients and an initial estimate of x.
Assign coefficients to the variables a, b, c, and d.

(13) In a script file you prompt the user who then inputs the value of each variable as in calc_roots.m where the first two lines of the algorithm were,…

Prompt the user for the coefficients a, b, and c
Input the coefficients a, b, and c.

(14) All of our algorithms will start with the input data.

Adding Information to a Plot.

(15) A graph, chart, or table of data isn't complete without information telling us everything we need to know about the data being displayed. You can't just show numbers and expect everyone to know what you mean.

(16) When making a plot in Matlab, you need to add three things.

A. A title - (what is the graph showing us?)
B. a legend - (what does each line represent?)
C. axis labels - (what values are being shown on the x and y axes?)

(17) In Matlab, we can add this information to our plots using the following commands.

>> xlabel (‘you put the correct words here’)

>> ylabel (‘again, you fill in what the y-axis is’)

>> title (‘This is where you type in the Title’)

>> legend (‘name for line1’, ‘name for line2’, name for line3’)

(18) You can also add titles, labels, and legends in the figure editor by clicking on the word Insert and then choosing either, title, x label, y label, or legened.

(19) And finally, there is an icon on the figure toolbar for adding a legend.

(20) I have found however the best way to add a legend is using the legend command from the Command Window prompt. This way, you type in directly what each line represents.

(21) Last of all, you must SAVE your edited figure! And don’t forget to save a copy of the final figure as a Windows bitmap (.bmp) file.

Using the Ones, Zeros, and Eye Functions

(22) To create a matrix that consists of all **ones** or **zeros**, we type in the follwing,…

>> A = ones(3,2) - makes as matrix of al ones with three rows and two columns.

>> B = zeros(5,3) - makes a matrix of all zeros with five rows and three columns.

(23) We can make both square and non-square matrices with these two commands.

(24) The **eye** command creates identity matrices, which by definition **must be square**.

>> C = eye(3) - makes a 3 X 3 identity matrix

Editing Matrices Using Subscripts

(25) When we were making the augmented matrices for Cramer's Rule, we edited the matrices in the array editor. A more direct way to edit a matrix is by using subscripts.

(26) To make a "augmented" matrix, we replaced one of the columns representing one of the unknowns (i.e. x, y, z, etc.) with the column vector b.

(27) We could have done this without using the array editor by typing,…

>> Aaug_x = A - sets the matrix Aaug_x = A

>> Aaug_x(:,1) = b - replaces the values in column one with the column vector b

(28) In the line of code above, the subscripts tell Matlab to replace the 1st column with the values in the vector b. What would happen if we had typed in the following?

>> Aaug_x(:, 2) = b replaces the_____with b

>> Aaug_x(2, :) = b replaces the_____with b

>> Aaug_x(:, 3) = b replaces the_____with b

(29) In lab this week, we will be writing a function that uses the ones command to create a matrix and then we will use subscripts to replace one of the columns with some data.

MAT files

(30) MAT files are the datafile format for Matlab. If I want to save all of the variables in the Matlab Workspace to a file called "asignment6", I type,…

>> save assignment6

which then creates a MAT file "assignment6.mat".

(31) To retrieve the data from the MAT file, I type,..

>> load assignment6

(32) Look up MAT files in your textbook for Wednesday.

Figure 1. Bank Balance Over Ten Years