APM 153 LECTURE ELEVEN - Iteration, For-Loops, Nested For-Loops, Subscripts Plotting in Matlab, Introduction to the Course Project.

Iteration

(1) "Iteration" is similar to "repetition" in that it refers to **an event** that is **repeated**.

(2) For example, some organisms are capable of reproducing **more than once** in their lifetimes. Organisms that have more than one set of offspring are said to be **iteroparous.**

(3) In contrast, many organisms, such as many insect species, reproduce only once. Organisms that reproduce once are said to be **semelparous.**

(4) In mathematics and computer science **iteration** also means **how many times** some calculation or operation is performed.

(5) For example**, the first time** we do an operation, we say we are going through the **first iteration.** The second time is the **second iteration**, and so on.

(6) Chapter Four of your textbook introduced two forms of iteration: **while loops** and **for-loops**. We used a **while loop** in root3.m to repeat the Newton-Raphson equation.

For-Loops

(7) The other form of iteration in Matlab is called a **for-loop** which in other programming languages is also known as a "**do-loop**". The word loop, of course means "to go around and around" and the word "do" means **do this operation**.

(8) The reason why this type of construct is called a for-loop in Matlab is because for-loops start with the word **for**.

(9) For-loops start with the word **for** followed by some **instructions** about how many times the loop is to repeat as in the example below.

```
for  i = 1: 100
        do this operation
end
```

(10) In English, the for-loop above says, "**for every value of the counter *i* from 1 to 100, do this operation**".

(11) In the for-loop above, the symbol *i* is used as a **counter** to keep track of what iteration the loop is currently on.

(12) The **first time** through the loop, the first value of *i* is 1. The second time through the loop, the value of *i* is **incremented** to 2. The third time through the loop the value of i is **incremented** to three.

(13) We use the word **incremented** to mean that the value has been increased by one unit or time step.

(14) In the for-loop above, the value of i is incremented from 1 to 100 which means that the loop will repeat 100 times. How would you re-write the for-loop above so that it repeats once a day for an entire year?

```
for i = 1: _____
        do this operation
end
```

(15) On the lines below, write a for-loop to perform an operation once a month for two years. Use the correct Matlab syntax.

_____

_____

_____

(16) You will notice that the stuff **inside** a for-loop is **indented**.

Nested For-Loops

(17) It is possible to write a for-loop inside of another for-loop. When we write one for-loop inside another the construct is called a **nested for-loop**.

(18) For example, let's imagine that we had a program which calculated the amount of money you owe on your credit card every day for twelve months. A nested for-loop to perform this calculation might look something like the following **pseudocode**.

```
number_of_days = [31, 28, 31, 30 31, 30, 31, 31, 30, 31, 30, 31]

for month = 1: 12
    for day = 1:  number_of_days(month)
        credit_due = credit_due + interest + new_charges – payments
    end
end
```

(19) In the example above, we needed some way to account for the fact that different months have a different number of days.

(20) One way to handle this problem is to set a vector called **number_of_days** equal to the number of days in each month.

(21) In the inner for-loop, each month the counter **day** is incremented from one to the correct number_of_days. How does the program know which value is the correct number_of_days?

Subscripts

(22) As the outer for-loop increments, the value of month increments from 1 to 12. In the first iteration, the value of month is equal to 1. Therefore, in the inner for-loop, the value of **day** increments from 1 to number_of_days(1), which is the value 31.

(23) Remember, the variable, number_of_days is a **vector with twelve elements** representing the number of days in each month. So, number_of_days(1) is the first element in the vector and that value is 31.

(24) When we use the vector number_of_days(1) or number_of_days(month) the information inside the parentheses is called the **subscripts**. Subscripts tells us **which element in the vector**.

(25) Because **subscripts** tell the program **which element** to use, they are like an **address**.

Introduction to Plotting in Matlab

(26) Subscripts help programs keep track of the elements in an array. As such, they are particularly useful for plotting data.

(27) For example, while writing your root3.m program, if you did not suppress echoing using the semicolon, you probably saw all the estimates of xnplus1 as they were generated.

(28) In the original version of root3, all we needed for output was the final value of xnplus1. But what if we wanted to save all the values of xnplus1 as they were generated?

(29) Make the following changes to your root3.m

(a) add the output argument "love"	-	function love = root3(coef, xn)
(b) above the while loop add	-	i = 0
(c) in the while loop add	-	i = i + 1
(d) on the line above the "end add	-	love(i) = xnplus1

(30) Now when you run your root3.m you can plot all the values of xnplus1 by typing

```
>> plot (root3(c, 1))
```

(31) To plot another line on the same graph, type **hold.**

(32) Then type `>> plot (root3(c,100), 'r+')`

(33) Next, type `>> plot (root3(c,-100), 'g+')`

APM 153 Spring 2006 Course Project

(1) Each student will design and carry out a course project using Matlab. The course project will count as one regular homework assignment.

(2) Each project will include a Matlab program written as either a script file or a function.

(3) Each project will be based upon a mathematical problem from one of the other courses you are taking now. Your project **must** relate to one of your current courses.

(4) The course project is an **opportunity** for each student to **apply** the **computer methods** we are learning in this course to their other courses

(5) There will be **three parts** to the project. The **first** part will the **PROPOSAL**. The proposal will consist of a one page description of what you will do for your project.

(6) Your proposal will tell me (a) what problem you are going to solve, (b) how this problem relates to one of your current courses, (c) what kind of input your problem will use, and (d) what kind of output your program will generate.

(7) Your proposal will be due in class on Monday, February 27th. The proposal is worth 10% of your course project. **Late proposals will be receive a grade of zero out of ten.**

(8) You will hand in your proposal as an MS Word document which your instructor will review and then hand back to you with any suggested changes on Monday, March 6th.

(9) The second part of your project will be a printout of your Matlab program (as a script file or a function), as well as your algorithm written as pseudocode and as a flowchart

(10) The second part of the course project will be **due in class on Monday, March 27th.**

(11) You will hand in a printout of your program or function and you will send an electronic copy to jcornell@esf.edu. The program will be worth 40% of the assignment. **Work handed in late will receive a zero out of forty**.

(12) Your program or function should include the all necessary documentation and a "help file" (both programs written as script files and functions can have help files).

(13) Remember, proper documentation includes,…

(1) Name, Date, Record of Revisions,
(2) Program Description
(3) Dictionary of Variables
(4) Section Headers
(5) Additional line comments where needed

(14) In addition, your program should do what it is supposed to do and should generate the correct output. What is more, your output should be complete and useful.

(15) By complete, your output should use complete sentences and any necessary units. For example, if your program calculates the amount of resistance across a circuit, the output should be something like,…

The total resistance in the circuit is 1.65 ohms.

(16) Although not absolutely necessary, due consideration will be given to projects that display output using the **plot** or **ezplot** functions.

(17) Once your instructor has evaluated your program and returned your hardcopy along with any suggestions, you will then finish your project by writing a short (3 – 4 page) MS Word document.

(18) Your final document should include,

Part One – An introduction describing what your program does, what output it generates and how your program relates to one of the classes you are taking now.

Part Two – your algorithm written as pseudocode and as a flowchart.

Part Three - a copy of your Matlab program

Part Four – a copy of your diary file showing that your program works correctly

Part Five – copies of any graphs or plots created as output.

(19) Your final paper will be due in class on Monday, April 24th.

(20) If you need help with any part of your project, come and see your instructor.

(21) You may also talk over your ideas with your classmates, but your course project must be your own work!

"If no one trusts your work, no one will trust you,… and vice versa."