

APM 153 LECTURE NINE Running root3.m, Functions, Review for First Exam.

Running root3.m

(1) To use your root3.m program, you need to give the program a vector containing the coefficients of a cubic equation, and an estimate of the value for one of the roots.

(2) Given the equation, $3x^3 + 4x^2 - 7x + 20 = 0$, you would set up a vector c so that,...

$$c = [3 \ 4 \ -7 \ 20]$$

(3) To find one of the roots, at the Matlab prompt, you would type in **root3(c,1)**.

(4) Once you run your root3 function for the first time, it will tell you what kinds of roots you are looking for. Knowing what kinds of roots you are looking for will help you find all three roots.

(5) You can check the results of your root3.m function by using Matlab's root finder **roots.m** by typing in the command `roots(c)`.

(6) Notice that **roots.m** finds all three roots at once and does not require an estimate

Functions vs. Script Files

(7) In Chapter 3 we read about **top-down design**. One part of top-down design is decomposition where complicated algorithms are broken down into smaller subprograms. In Chapter 5 we read how functions can be used to create subprograms.

(8) Writing a function is like creating a new button on your calculator. The functions **cos**, **sin**, and **sqrt**, are all Matlab functions that work just like the buttons with the same name on your scientific calculator.

(9) How do we use the buttons on our calculator? Typically, first we enter a number, and then press the desired button. Or, on your calculator, you press the button then enter the number.

(10) Either way, when we press the button, say COS, the number is “passed” to the function where it is processed and “returns” the answer to the calculator’s screen.

(11) In Matlab functions pass input and output data as “lists of arguments”. In our calculator example the original number we entered was our *input argument* and the answer was the *output argument*.

(12) Functions are written as M-files. Up to now, the M-files we have seen have been Script Files which perform a series of commands just as if they had been typed into the Matlab Command Window.

(13) Script files **share** the Command Window’s Workspace so that any variable created in the script file remain in the Workspace window after the Script file is run.

(14) Functions are different. Functions run in their own independent workspace and they only share variables listed as input and output arguments.

(15) Functions therefore are **isolated** from the rest of Matlab, which is one of the characteristics that make functions ideal for writing subprograms.

(16) In addition, functions can be used to create reusable pieces of code. If you write a function to calculate a particular problem, you can reuse that function in any program that requires a similar solution.

(17) In summary, functions

- create independent pieces of code
- create reusable code
- allow big programs to be broken down into subtasks
- once tested, should always work (unit testing)

Review for Exam

(18) Don’t forget to bring your textbook, 3-ring binder, and USB drive to the exam.

(19) Everything covered up to and including today’s lecture will be on the exam.

(20) Cooperation between students is not allowed during the exam.

(21) If you have a problem with the test or your computer, ask the instructor or a TA.