

CS 105 – Introduction to Scientific Computing

Assignment 9 – Scripts with Loops II

Objectives

After completing this assignment you should be able to:

1. Write scripts with loops to solve these problems

Overview

Algorithmic problem solving is a critical skill for scientific computing. Part of this process involves translating the problem to use the language of scientific computing. This includes identifying where to do I/O and when to use branches and loops. This process is similar to that of solving word problems.

In solving the problems of Part III you should consider attempting to reword the problem to use the language of scientific computing, that is to create *pseudocode*. From pseudocode it should then be relatively easy to convert this to Matlab code.

Part I: Textbook Exercises

Section 4.7

- #4.25
 - NOTE: You may assume that the user types valid inputs

Part II: Script Exercises

1. Create a vector X as a 1x8 random vector. Write MATLAB code that displays the maximum and minimum values in the vector. Your code should work for any vector of length > 0 .
 - a. Do this using only MATLAB functions and no loops.
 - b. Do this using one `for` loop and no MATLAB functions (other than length).
 - c. Do this using one `while` loop and no MATLAB functions (other than length).
2. Create a vector X as a 1x8 random vector. Write MATLAB code that displays the mean of the values in the vector. Your code should work for any vector of length > 0 .
 - a. Do this using only MATLAB functions and no loops.
 - b. Do this using a loop and no MATLAB functions (other than length). You choose the type of loop.
3. Here is a tried-and-true method for calculating the square root of a positive number C using only basic arithmetic operations. The method uses two variables, X_{old} and X_{new} , and updates the variables repeatedly until the absolute value of the difference $X_{old} - X_{new}$ is less than some threshold; for this question, the threshold is 0.0001. The rules for updating the variables are as follows:
 - a. Before the loop, assign $X_{old} = C$
 - b. Loop until the difference is below the threshold, performing these two calculations in this order:

$$X_{new} = X_{old} - \frac{X_{old}^2 - C}{2X_{old}}$$
$$X_{old} = X_{new}$$

The difference $|X_{old} - X_{new}|$ is tested **between** these two calculations. With each successive iteration, the difference becomes less than the time before. Once the difference becomes less than the threshold, the loop should terminate and the answer (i.e., the square root of C) is in X_{new} .

Implement this algorithm in MATLAB, displaying the final value. Verify that your code produces the correct answer by comparing your answer to that produced by MATLAB's `sqrt` function on a few test values. Allow the user to enter C . You may assume they enter this as a valid positive number.

HINT: One way to solve this is to have an *infinite loop*, that is to force the loop to go on forever and only end via a *break* statement

Submission

Submit a single zip file consisting of

- Your report, which will contain your thoughts/experiences for work done in Parts I and II
- Your scripts, one per problem, for Parts I and II