

Homework 3

Goal: To learn to write Matlab scripts to create simple programs using loops and conditional statements.

Due: March 5

Read and work through the calculations in Spencer and Ware, Chapters 4 and 7. Explore the Programming Basics examples on Matlab-Monkey.com. I recommend downloading the full set of examples to your computer.

Part A: Hand-written Problems - To be turned in on paper

1. (10 points) Describe the results of each of the following Matlab statements. Try to figure them out on your own first. You can then check your results by typing each command into Matlab.

(a) `A = 1:6`

(b) `B = linspace(0,2,100)`

(c) `length(A)`

(d) `length(B)`

(e) `A(2)`

(f) `B(2)`

(g) `A(3:4)`

(h) `B(end)`

(i) `C = [1 3; -1 4]`

(j) `C(:,1)`

2. (12 points) Write some Matlab code to calculate each of the following. Assuming `x` and `y` are both row vectors with the same dimensions.

(a)

$$z = \frac{1}{x}$$

(b)

$$z = \frac{xy}{x+y}$$

(c)

$$z = 4e^{-(x^2+y^2)/2a}$$

3. (5 points) The following sample code contains a for loop. Examine the code without running it! Make a table by hand to show the values of `k` and `a` on each iteration of the loop. What is the final value of `a`? Check your answers by running the code.

```
a = 1;
for k=1:5
    a = a * k;
end
a
```

4. (5 points) The following sample code contains a for loop. Examine the code without running it! Make a table by hand to show the values of **k** and **a** on each iteration of the loop. What is the final value of **a**? Check your answers by running the code.

```
x = 0;
for k=1:10
    if k < 5
        x = x + 1;
    elseif k == 5
        x = x * 2;
    else
        x = x - 3;
    end
end
x
```

5. (5 points) Write some Matlab code to test to see if each element of a row matrix **A** is positive, negative or zero. Create a new matrix **B** that equals 1 if **A** > 0, that equals -1 if **A** < 0 and that equals 0 if **A** = 0.

Part B: Coding Problems - To be turned in on Blackboard

1. (20 points) *Divergent Series* The series

$$\sum_{n=1}^N \frac{1}{n}$$

diverges very slowly as N approaches infinity. Calculate the sum using both a for loop and the Matlab function `sum()`. Create a nicely formatted table (with numbers lined up in columns and descriptive labels for each column) showing the value of the sum for the following values of N : 10, 10^3 , 10^5 , 10^7 , 10^9 . In your table show the value N , the sum, computation time using for loop, and computation time using sum command. The last calculation with a billion computations might take a while (perhaps a couple minutes). If your code gets stuck, type ctrl-C in the command window. This problem is related to matlab-monkey programs A02.m, A04.m, A05.m and A06.m.

2. (20 points) *Taylor Series Approximation*. Starting with example programs A05.m, A07.m and/or A09.m write a program to calculate the Taylor series approximation of the sine function:

$$\sin(x) = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!}$$

The program should prompt the user to enter the value x and the desired tolerance ϵ_{\max} . The program should then determine the fewest number of terms in the series required to approximate the function to the desired tolerance. I.e. if $f(x)$ is the function and $f_{\text{approx}}(x)$ is the Taylor series, you want to include just enough terms in the series so that $|f_{\text{approx}}(x) - f(x)| < \epsilon_{\max}$. Your program should print out the number of terms needed to achieve the desired tolerance, the approximate value $f(x)$ using the Taylor series $f_{\text{approx}}(x)$ and the error. Run your program for the following cases:

- $\sin(x)$ at $x = 0.1$ radians with a tolerance of 10^{-3} .
 - $\sin(x)$ at $x = 7\pi/4$ with a tolerance of 10^{-3} .
 - $\sin(x)$ at $x = 7\pi/4$ with a tolerance of 10^{-6} .
3. (20 points) *Guessing Game*. Write a guessing game program. Generate a random integer between 1 and 100. This will be the target number the user tries to guess. Prompt the user to pick an integer

between 1 and 100. Tell the user if their guess is “cold”, “warm”, or “you won” (in the unlikely case that they guessed it correctly). Then prompt the user to enter a second guess and tell them if they are “warmer” (if their next guess is closer to the target) or “colder” (if it is farther from the target). Keep going until they guess the correct number. Keep a record of how many times they guess and display that value when the game is over. Feel free to write some code that comments on how well they did. Extra Credit: what is the best strategy to win this game?