<u>Math 266A Section 01 (Barsamian) MATLAB 2: Sequences</u>

In Chapter 2 of the textbook, you were introduced to sequences. These are simply functions whose domain is the set $\mathbb{N}=\{0,1,2,3,\cdots\}$. The outputs of such a function can be thought of as a list of numbers. Equivalently, every list of numbers can be thought of as the output of a sequence. The first sequences that you encountered were *explicitly defined* as a function of *n*. But you also learned about *recursively defined* sequences. For these, the first number on the list is given, and you are given a formula that tells you how to produce, given any number on the list, the next number on the list. In this MATLAB exercise, you will explore both *explicitly defined* and *recursively defined* sequences. In the course of doing the exercise, you will learn a few new basic MATLAB skills, including:

- defining *arrays* of numbers
- graphing a function whose domain is an *array* of numbers
- changing the line style of a graph to indicate a discrete data set
- telling MATLAB to display more digits
- using the *up arrow* and *down arrow* keys as a typing shortcut
- using MATLAB to produce the outputs of a *recursively defined* sequence

In exercise 2.1#2 from Homework 2, you are asked to make an input/output table and graph for the function $f(n)=\frac{1}{(1+n^2)}$, where $n=0,1,2,3,\cdots$. Observe that f is really a sequence, because its domain is the set $\mathbb{N}=\{0,1,2,3,\cdots\}$. Your first chore is to use MATLAB to find the values for the table and to make the graph. You will start by creating an *array* of *n*-values.

1) Start the MATLAB program. You should see the *command window*, with the command prompt `>>`.
2) Type `>>n=0:1:10`[Enter] This creates a list of numbers starting at 0, spaced 1 unit apart, and ending at 10. Such a list is called an *array*; you have just told MATLAB to create an *array* and to call it *n*. MATLAB will respond by displaying the list of numbers in the *array n*.

Next, you will create an *array* of *y*-values.

3) Type `>>y=1./(1+n.^2)`[Enter] Be sure to include the periods. This command tells MATLAB to create an array, called *y*, whose elements are obtained by performing the calculation $1/(1+n^2)$ on every number in the *array* called *n*. Notice that you were told to type `./` and `.^` instead of just `/` and `^`. This was because *n* is an *array*: the command had to be typed using these special *array arithmetic* operations.
4) MATLAB will respond by displaying the list of numbers in the *array y*. Write down the first five numbers on this list. They are the five *y*-values corresponding to the *n*-values 0, 1, 2, 3, 4.
5) Use the numbers that you have written down to produce the table called for in exercise 2.1#2.

Next, you will plot the data that you just created.

6) Type `>>plot(n,y)`[Enter] This command tells MATLAB to create a plot of ordered pairs of numbers, where the left number of each pair is one of the numbers in *array n*, and the right number of each pair is the corresponding number in *array y*.

The graph will pop-up in a new window entitled *Figure 1*, or something similar. The graph will not look very smooth, because it was produced using only eleven data points. But remember that the goal was to produce a graph for a function whose domain is the set $\mathbb{N}=\{0,1,2,3,\cdots\}$. Such a graph should really just be a bunch of dots not connected by a curve. You will now edit your graph to make it look that way.

7) In the window containing your graph, find the *figure toolbar* near the top of the window, find the white arrow in that toolbar, and click on the white arrow. Clicking this arrow switches MATLAB in and out of *Edit Plot* mode.

8) Using the arrow, double-click somewhere on the axes of the graph. This will open up a smaller window called *Property Editor-Axes*.
9) Using the *Property Editor-Axes* window, set the limits of the axes so that the *x*-axis goes from 0 to 10 and the *y*-axis goes from 0 to 1. (If necessary, refer to MATLAB 1 for review.)
10) Using the arrow, click somewhere on the curve of the graph, itself. The smaller window should change to one called *Property Editor-Lineseries*.
11) Click on the little downward-pointing triangle next to the box called *Line*. This will open up a short list of line styles.
12) From the list of available line styles, choose *no line*.
13) Click on the little downward-pointing triangle next to the box immediately to the right of the word *Marker*. This will open up a list of marker styles.
14) From the list of available marker styles, choose the solid dot marker.
15) Click on the little downward-pointing triangle next to the box that is to the right of the box that you just used. This will open up a list of marker sizes.
16) From the list of available marker sizes, choose marker size 15.0.
17) Add text notes to the graph using steps similar to those you used when adding notes to your graph in MATLAB 1. Be sure to include the function description, the exercise number, your name, and the date.
18) Click again on the white arrow to tell MATLAB that you want to get out of *Edit Plot* mode.
19) Print your graph.

In exercise 2.2#96 from Homework 2, you are asked to find the fixed points of a sequence defined by the recursive formula $a_{n+1} = \left(-\frac{1}{3}\right)a_n + \left(\frac{1}{4}\right)$. You don't need MATLAB to find the fixed points: a simple algebraic calculation shows that there is a single fixed point at the value $a = \frac{3}{16} = 0.1875$. But MATLAB does give you a very useful way of observing the *convergence* of the sequence values to the fixed point.

20) Type `>>clear`[Enter] to clear the MATLAB memory.
21) Type `>>syms x`[Enter] to create a symbolic variable called *x*.
22) Type `>>y=(-x/3)+(1/4)`[Enter] to create a function called *y*.

Compare the function that you have just created to the one described in exercise 2.2#96. In the exercise, the equation $a_{n+1} = \left(-\frac{1}{3}\right)a_n + \left(\frac{1}{4}\right)$ describes a function whose input is a_n and whose output is a_{n+1}. The function that you just typed into MATLAB is identical, except that the input is called *x* and the output is called *y*. In the next ten or so steps, you will familiarize yourself with different ways of feeding input into the function and using the resulting output. You will start with the simplest method: substituting a number into the function.

23) Type `>>subs(y,3)`[Enter] MATLAB should respond by displaying `ans = -.75`, because $\left(-\frac{1}{3}\right)\cdot 3 + \left(\frac{1}{4}\right) = -1 + \left(\frac{1}{4}\right) = -.75$.

Now you will substitute a number in to the function *y* by first setting *x* equal to the number and then substituting *x* into *y*.

24) Type `>>x=7`[Enter] to tell MATLAB to set the value of the variable *x* to 7.
25) Type `>>subs(y,x)`[Enter] to tell MATLAB to substitute the current value of the variable named *x* into the function named *y*. MATLAB should respond by displaying `ans = -2.0833`, because $\left(-\frac{1}{3}\right)\cdot 7 + \left(\frac{1}{4}\right) = \frac{-25}{12} \approx -2.0833$.
26) Type `>>format long`[Enter] to tell MATLAB to display more digits.

In the next three steps, you will learn how to use the *up arrow* and *down arrow* keys as a typing shortcut.

27) Hit the *up arrow* key on the keyboard. MATLAB should respond by displaying the most recently entered command next to the command prompt. MATLAB keeps a record of all of the commands that you type at

the command prompt. Each time you press the *up arrow* or *down arrow* keys, MATLAB will move up or down the list of stored commands.

28) Using the *up arrow* and *down arrow* keys, go up and down the list of stored commands until you see the command `>>subs(y,x)`.
29) Hit `Enter`. MATLAB should respond with `ans = -2.08333333333333`. (Notice all the digits!)

So far, you have simply fed numbers into the function *y* and observed the resulting output. Now, you will use the function *recursively*. That is, each time you obtain an output, that output will be re-used as an input in order to produce yet another output. The result will be a list of numbers.

30) Type `>>x=3` `Enter` to tell MATLAB to reset the value of the variable *x* to 3.
31) Type `>>x=subs(y,x)` `Enter` This command tells MATLAB to substitute the current value of the variable *x* (that is, the number 3) into the function *y*, and then store the resulting number as the new value of the variable *x*. MATLAB should respond by displaying `x = -.75000000`.
32) Now press the *up arrow* key once, so that MATLAB displays the line `>>x=subs(y,x)`.
33) Hit `Enter` to tell MATLAB to execute the command. This command tells MATLAB to substitute the current value of the variable *x* (that is, the number -.75) into the function *y*, and then store the resulting number as the new value of the variable *x*. MATLAB should respond by displaying `x = 0.5`, because $\left(-\frac{1}{3}\right)\cdot(-.75)+\left(\frac{1}{4}\right)=.25+.25=.5$.

Realize that you have been building a list of numbers. The first number on the list is $x=3$, the second number is $x=-0.75$, the third number on the list is $x=0.5$. Write these numbers in a list.

34) Continue entering the `x=subs(y,x)` command in order to lengthen your list of numbers. (Use the *up arrow* and *down arrow* keys to save you the effort of re-typing the command each time.) Each time you use the command to obtain a new number for the list, write down the number. Do this until your list contains ten numbers. For your sanity, you need only record the first five decimal digits of each number.

Observe that the numbers on your list are getting closer and closer to the value 0.1875. It was mentioned above that an algebraic solution of exercise 2.2#96 finds a fixed point of $a=\frac{3}{16}=0.1875$. It seems that the numbers on the list that you just created are getting closer and closer to this value. This behavior is called *convergence*.

Now you will build a second list of numbers.

35) Type `>>x=0.1875` `Enter` to tell MATLAB to store the value 0.1875 in the variable named *x*. This is the first number on the list. Write this number down.
36) Continue entering the `x=subs(y,x)` command in order to lengthen your list of numbers. Each time you use the command to obtain a new number for the list, write down the number. Do this until your list contains five numbers. You need only record the first five decimal digits of each number.
37) Why does this new list come out the way it does? (Write down your answer to this question.)

Now you will build a third list of numbers.

38) Reset the variable *x* to a number of your choosing. To get credit for this part of the assignment, you must choose an *x*-value that nobody else in the class has chosen.
39) Use the `x=subs(y,x)` command repeatedly in order to lengthen your list of numbers. Each time you use the command to obtain a new number for the list, write down the number. Do this until your list contains ten numbers. You need only record the first five decimal digits of each number.

In addition to the printed graph for exercise 2.1#2, you should have another page (either hand-written or type-written) on which you have written the three lists of numbers created and your answer to the question posed in step number (37). Make sure that this additional page is titled with your name, the MATLAB assignment number, and the date. Turn all this stuff in with your Homework #2.