MATLAB 6: Supporting Work for 5.4#22 and 5.6#16

In this final MATLAB exercise, you will use three old MATLAB skills:

- differentiating functions (done in MATLAB 1)
- making graphs with specific points highlighted (first done in MATLAB 4)
- producing output of recursively-defined functions (done in MATLAB 2)

In addition, you will learn some new skills:

- Creating functions of more than one variable
- Differentiating a function of more than one variable with respect to a particular variable
- Solving an equation
- Taking higher derivatives
- Using MATLAB to assist in performing the second derivative test.

In exercise 5.4#22a from Homework 6, you are asked to find the value of *x* that maximizes the function $r(x) = \frac{\ln\left[e^{-ax} \cdot bx^c\right]}{x}$, where *a*, *b*, and *c* are positive constants. That is a very hard exercise, but you should try to do it by hand. Your first MATLAB chore will be to check your work by using MATLAB to do the problem

1) Start the MATLAB program. You should see the *command window*, with the command prompt `>>`.

2) Type `clear` `Enter`

3) Type `syms a b c x` `Enter` Be sure to include the spaces.

4) Type `r =log(exp(-a*x)*b*x^c)/x` `Enter` This creates the function $r(x)$. The symbols *a*, *b*, and *c* are constants.

5) Type `Dr=diff(r,'x')` `Enter` This differentiates $r(x)$ with respect to *x* and calls the result *Dr*.

6) Type `simplify(Dr)` `Enter` in order to see a simplified version of *Dr*.

7) Question: what does MATLAB give as an answer?

8) Question: does the *Dr* produced by MATLAB agree with the $r'(x)$ that you got in your solution to 5.4#22a? (If your answer is “no”, then go back and check the work that gave you your $r'(x)$.)

9) Type `xcrit=solve(Dr)` `Enter` This solves the equation $r'(x) = 0$. (Remember that an *x*-value that causes the derivative to be zero is called a *critical number*. That is why I chose the name *xcrit*.)

10) Question: what does MATLAB give as an answer?

11) Question: does the *xcrit* produced by MATLAB agree with the solution to the equation $r'(x) = 0$ that you got in your solution to 5.4#22a? (If your answer is “no”, then you should go back and check the work that gave you your solution to the equation $r'(x) = 0$.)

We now have a critical number for $r(x)$, but we do not know whether the critical number produces a local max in $r(x)$, a local min, or neither. To determine that, we need to do the second derivative test.

12) Type `DDr=diff(Dr,'x')` `Enter` to find the second derivative.

13) Type `simplify(DDr)` `Enter` to simplify it.

Now that we have the second derivative, we need to use the critical number as input to this function and examine the resulting output to see if the output is positive or negative. That will tell us if the function $r(x)$ is concave up (positive concavity) or down (negative concavity). (This is the second derivative test.) We denote the output by the word *concavity*.

14) Type `concavity=subs(DDr,x,xcrit)` `Enter` The result will be a horrible mess.

15) Type `simplify(concavity)` `Enter` The result will still be a mess.

The problem is that the expression for concavity involves the letters a, b, and c. We know that these are positive constants. With that information, we could look at the expression for concavity and (with a lot of hard thinking) figure out whether concavity is positive, negative or zero. But MATLAB is not smart enough to do that. We have to just give MATLAB actual values for *a*, *b*, and *c*, and let it give us a numerical answer for the concavity.

16) Type `subs(concavity,{a,b,c},{0.1,4,0.9})` `Enter` (These are the numbers given in the problem statement.)

17) Question: What answer does MATLAB give?

18) Question: Using the second derivative test, is $x = xcrit$ a relative max, a relative min, or neither?

We now turn our attention to the graph required for 5.4#22b. For it, we need to substitute actual values in for *a*, *b*, and *c*.

19) Type `r= subs(r,{a,b,c},{0.1,4,0.9})` `Enter`

20) Type `simplify(r)` `Enter` The result will be a function involving the variable *x* alone.

21) Type `xcrit= subs(xcrit,{b,c},{4,0.9})` `Enter` MATLAB should respond by giving a numerical value for *xcrit*.

22) Type `ycrit=subs(r,x,xcrit)` `Enter` MATLAB should respond by giving a numerical value for *ycrit*.

23) Question: Based on the previous two answers, what are the (x, y) coordinates of the relative max?

24) Type `X=0:0.01:5` `Enter` to create an array of *x*-values to be used as inputs to the function $r(x)$.

25) Type `R=subs(r,X)`[Enter] to create an array of the corresponding outputs.

26) Type `plot(X,R,'y',xcrit,ycrit)`[Enter] This will graph the function $r(x)$ in yellow and the point $(xcrit, ycrit)$ in black. This will enable you to see the single point.

27) Change the scale on the *y*-axis to show $0 \le y \le 1.5$. You may have to review an old MATLAB to recall how to do this.

28) Change the single point $(xcrit, ycrit)$ so that it is highlighted by an appropriately-sized black dot. (Those of you who have turned in MATLAB with crazy-big dots in the past have this last chance to figure out how to make a nice-sized dot!)

29) Change the graph of $r(x)$ so that it is a black line, thick enough to be easily seen but not so thick that it obscures the black dot.

30) Give the graph the following title, centered in a box with no lines

Graph of the function r(x)=log(exp(-a*x)*b*x^c)/x
using a=0.1, b=4, c=0.9
Your Name, Math 266A Homework 6
Due Friday March 10, 2006

31) Create the following comment in a box with no lines

The max occurs at xcrit=e/(b^{1/c})

32) Create an arrow pointing from the comment that you just created to the black dot.

33) Print this graph. Attach the graph and your answers to MATLAB questions 7), 8), 10), 11), 17), 18), and 23) to your solution of exercise 5.4#22.

Finally (!!) we turn our attention to exercise 5.6#16 from Homework 6. In that exercise, you are asked to consider the convergence of sequences defined by the recursive formula $N_{t+1} = 1.5N_t e^{-0.001N_t}$ for $t = 0,1,2,\cdots$. You don't need MATLAB to find the fixed points: an algebraic calculation shows that there are two fixed points. One is at at the value $N = 0$ and the other is at $N = \frac{\ln(1/1.5)}{.001} \approx 405.46$. But MATLAB does give you a very useful way of observing the *convergence* of the sequence values to the fixed point.

34) Type `>>clear`[Enter]

35) Type `>>syms x`[Enter] to create a symbolic variable called *x*.

36) Type `>>y=1.5*x*exp(-.001*x)`[Enter] to create a function called *y*.

Compare the function that you have just created to the one described in exercise 5.6#16. In the exercise, the equation $N_{t+1} = 1.5N_t e^{-0.001N_t}$ describes a function whose input is N_t and whose output is N_{t+1}. The function that you just typed into MATLAB is identical, except that the input is called *x* and the output is called *y*. 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.

37) Type `>>x=100`[Enter] to tell MATLAB to reset the value of the variable *x* to 100.

38) 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 = 135.7256`.

39) Now press the *up arrow* key once, so that MATLAB displays the line `>>x=subs(y,x)`

40) 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 135.7256) into the function *y*, and then store the resulting number as the new value of the variable *x*. MATLAB should respond by displaying `x = 177.7494`

Realize that you have been building a list of numbers. The first number on the list is $x = 100$, the second number is $x = 135.7256$, the third number on the list is $x = 177.7494$. Write these numbers in a list.

41) 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 21 numbers (corresponding to the values $t = 0$ through $t = 20$.) For your sanity, you need only record the first four decimal digits of each number.

Observe that the numbers on your list are getting closer and closer to some value. It was mentioned above that an algebraic solution of exercise 5.6#16 finds fixed points of $N = 0$ and $N = \frac{\ln(1/1.5)}{.001} \approx 405.46$. It seems that the numbers on the list that you just created are getting closer and closer to one of these values. This behavior is called *convergence*.

42) Question: What number does your list seem to be converging to?

Now you will build a second list of numbers.

43) Type `>>x=800`[Enter] to tell MATLAB to store the value 800 in the variable named *x*. This is the first number on the list. Write this number down.

44) 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 21 numbers. You need only record the first four decimal digits of each number.

45) Question: What number does this new list seem to be converging to?

46) You should have two lists written on paper, along with your answers to questions 42) and 45). Attach this paper to your solution to exercise 5.6#16.