Sizhuo(Cindy) Liu
Math 128A
September 18th, 2018

Programming Assignment 1 Write-Up

Part One: Write a program to take 6 numbers a, b, c, d, e, f and find the locations x_{min} and x_{max} of the absolute extrema of the third-degree polynomial function:

$$p(x) = cx^3 + dx^2 + ex + f$$

over the interval [a, b]

Steps taken:

1. Create a function of the form function rslt = polyMinMax (a, b, c, d, e, f)
   (Details: polyMinMax.m)
   - Find the derivative of the original input polynomial using the MATLAB built-in function polyder ( ).
   - Find the critical points of the polynomial by calculating the roots of the derivative
     - Use the appropriate quadratic formula covered in lecture 5 to avoid unnecessary cancelation of digits:

$$x = \frac{2c}{-b \pm \sqrt{b^2 - 4ac}}.$$

   - Filter out critical points that are outside of the range of consideration [a, b] using an if statement within a for loop
   - Evaluate the corresponding y values of the critical points, the starting point, and the ending point by plugging x values to the original polynomial equation
   - Find the minimum and maximum x values and their corresponding y values using the MATLAB built in function max( ): these are the absolute extrema for the polynomial
   - Set the output to be a vector: rslt = [x_{min}, x_{max}, $p(x_{min})$, $p(x_{max})$]

2. Report the extrema returned by the code in part1.m
   - The function was ran for the following cases:

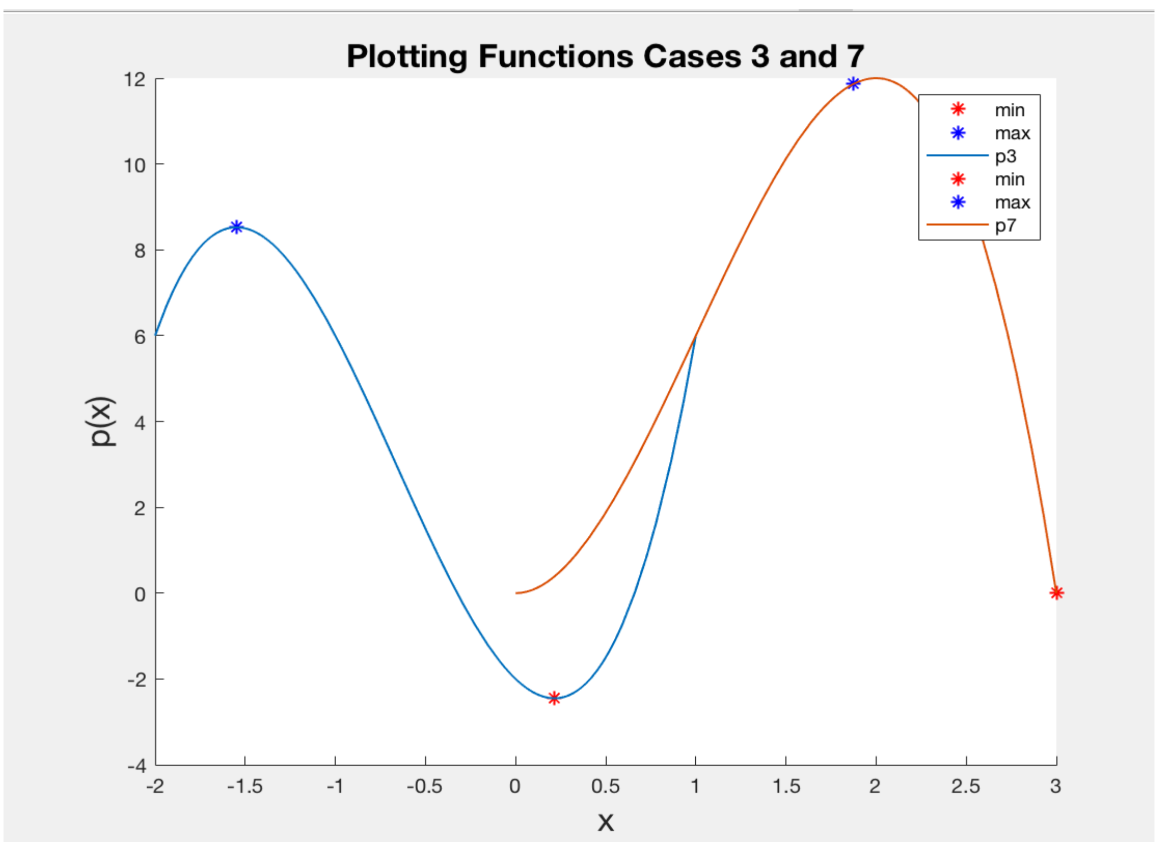
trial	a	b	c	d	e	f
1	-1	2	-1	2	-1	1
2	1	2	1	-2	-1	1
3	-2	1	4	8	-4	-2
4	-1	2	1	0	1	-3
5	-0.3	0.6	10^{-14}	9	-3	0
6	-1	2	0	0	0	1.7
7	0	3	-3	9	-10^{-14}	0
8	0	1	0	-2	3	-1

- The output is as following:

```
Command Window

>> part1
          2          -1          -1           5
    1.54858           1    -1.63113          -1
    0.21525    -1.54858    -2.45045     8.52452
         -0           2          -3           7
   0.166667        -0.3       -0.25        1.71
         -1          -1         1.7         1.7
          3      1.8765      -3e-14     11.8684
          0        0.75          -1       0.125
fx >>
```

- In addition, plots of p(x) over [a, b] for cases 3 and 7 was created through MATLAB's built-in function fplot ( ), with markers on the extrema
(Details: generate_plot.m)



Part Two: Write a code that takes an integer n and returns the nth term in the following sequence

$$a_1 = 1, \quad a_2 = \sqrt{1+2}, \quad a_3 = \sqrt{1+2\sqrt{1+3}}, \quad a_4 = \sqrt{1+2\sqrt{1+3\sqrt{1+4}}}, \quad \ldots$$

Steps taken:

1. Create a function of the form function a = nested_sqrt (n)
   (Details: nested_sqrt.m)
   - Set up a for loop, an if statement within the for loop, and another for loop inside the if statement so that the function eventually returns the value of the nth term
2. Evaluate a_n for $1 \leq n \leq 40$
   - Although it is possible to evaluate all 40 a_n with the nested_sqrt (n) function, the process is very slow and inefficient. Therefore, another function of the form

     function stored_data = an_stored (n)

     (Details: an_stored.m)

     is created (implementing nested_sqrt( ) as part of the function) in which an input value of n = 40 will return all values of a_n stored in a vector, shown as the following:

```
>> an_stored(40)

ans =

  Columns 1 through 11

    1.4142    1.7321    2.2361    2.5598    2.7551    2.8671    2.9292    2.9627    2.9806    2.9899    2.9948

  Columns 12 through 22

    2.9973    2.9986    2.9993    2.9996    2.9998    2.9999    3.0000    3.0000    3.0000    3.0000    3.0000

  Columns 23 through 33

    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000

  Columns 34 through 40

    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000    3.0000
```

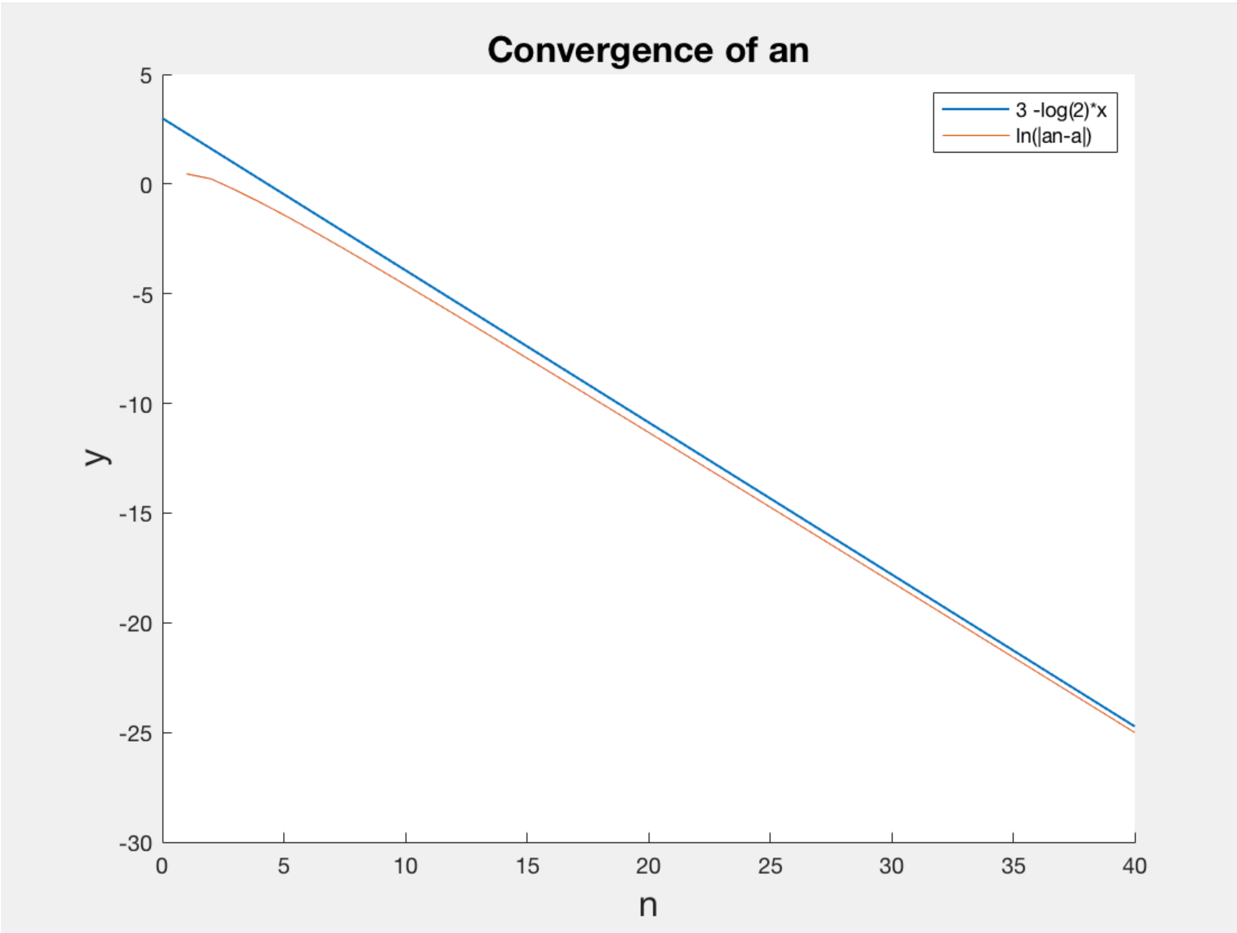
3. Make a plot of $\ln(|a_n - a|)$ vs n and as well as the line $y = 3 - \ln(2)n$
   (Details: plot_convergence.m)
   - From the vector of 40 values of a_n for $1 \leq n \leq 40$, it is obvious that the limiting value of the sequence

     $$a = \lim_{n\to\infty} a_n,$$

     is 3.0000.
   - Method for plotting the line $y = 3 - \ln(2)n$ is the same as for part one (using fplot ( ))
   - To plot $\ln(|a_n - a|)$ vs n, set n to be numbers 1 to 40 and set y to be $\ln(|a_n - a|)$ where a_n is the set of data obtained by calling the function an_stored (40) and a is 3, the limiting value of the

sequence guessed earlier.

- The combined plot is as the following:



- From the plot, the sequence βn would be $e^{3-\ln(2)n}$ as calculated by the following steps:

$$\lim_{n\to\infty}\left(\ln\left|a_n - a\right|\right) = 3 - \ln(2)n$$

$$\ln\left|a_n - a\right| \leq 3 - \ln(2)n$$

$$\left|a_n - a\right| \leq e^{3-\ln(2)n}$$

$$a_n - a = O\left(e^{3-\ln(2)n}\right)$$

$$\beta_n = e^{3-\ln(2)n}$$