Homework 1 – Report

John Wu

31 January 2020

1 Executive Summary

The class was tasked with creating a program that utilized Horner's algorithm as discussed in class to evaluate a polynomial and compute a derivative. This report outlines my understanding of the math and the method at which I implemented said algorithm.

2 The Math

Horner's algorithm is quite simple really. It describes a process in which one expands a polynomial into coefficients and powers of x. Simply put, you receive your polynomial and separate its constant, if any, and pull out an x from the remainder of the equation. You then repeat this task until you reach the leading coefficient.

Example 1:

$$\boldsymbol{f(x) = 3x^3 + 10x - 18 \rightarrow -18 + x(10 + 3x^2) \rightarrow -18 + x\big(10 + x(3x)\big) \rightarrow -18 + x\Big(10 + x\big(0 + x(3)\big)\Big)}$$

3 Implementation

In my implementation I used the variables that were given to us: a, d, x, y. I used y as an output variable, storing each iteration of my loops within it, displaying it at the end. The variable a was used to store the vector or list of coefficients including 0's, d was used to determine whether the code was to evaluate the polynomial (0) or compute the derivative (1), and finally x was used to determine what point I wanted to evaluate/derive at.

Evaluation: Here I took advantage of Horner's algorithm by taking the innermost number and working my way towards the outside. I started by taking the very first value given to me by my vector (a(1)), and setting y equal to it. Going from there, after each iteration I would multiply my **y** variable by **x** and add the next number in the vector. Thus, using the example shown before starting from the innermost part of the expanded polynomial, I would receive a 3 for my first y value. I would then proceed to multiply it by whatever I set x to (let's say 8) and add 10, I would receive a new y value of 34.

Derivative: Here things were a little trickier, instead of working from the inside out like I'd originally wanted to, we worked our way inside quite intuitively. We start with the last variable in our a variable, this is the constant of our polynomial if any exists and thus must be 0 after derivation. We then move to the next to last value given by our a variable, multiply it by x, derive, and plug in our x value. We continue this trend, multiplying each variable by x^i until we reach the end, thus we reach our answer.

4 Result

The results show up to be consistent throughout. I tested multitudes of values, hand checking each set of polynomials and utilizing multiple online resources to ensure their accuracy. The code works as intended provided the a vector holds all coefficient values for all relevant degrees of x. As shown in the tests below:

Test 1: vector a = [50 -2 80 0 1000 0 0 220 0 0 0], x = 42

- **Evaluation w/ code:** 8.540*10^17
- **Evaluation w/ calculator:** 8.5397*10^17
- **Remarks:** As it is difficult to ascertain whether the program simply rounded 8.5397 to 8.540 due to rounding 7 up causes 9 to also round up, it is up in the air as to whether this is a source of error or not. To further ascertain this, I computed a few polynomials of lower magnitude.

Test 2: vector a = [50 0 -2 80 1000], x = 42

- **Evaluation w/ code:** 155,585,632
- **Evaluation w/ calc:** 155,585,632
- **Derivative w/ code:** 14,817,512
- **Derivative w/ online calc:** 14,817,512
- **Remarks:** As shown in test 2, the results are consistent with that of other methods. I think it is worth noting further that when checking the validity of the derivative function, I checked and compared with three programs/websites links shown below.

5 Conclusion

The results shown above indicate that the program I created is, at the very least, *fairly* accurate. However, I do think that more varied tests would come to a more conclusive answer. I think that the discrepancy between results in test 1 are largely due to rounding, though I know that it would be foolish to say that the code is 100% accurate. Due to issues with the sizes allowed to integers in my program, it is reasonable to assume that there is error in the larger numbers, but they likely begin far past what is represented by Matlab. The results I've found do make me wonder about how accurate the calculations are exactly. The part I've struggled with most in this assignment was the derivative portion. At first, I was puzzled at how I was to approach this situation following the evaluation method I posed above, but after taking a step back and about 30 minutes mulling it over, I realized I was forcing a rather inelegant method than what else was immediately available.

Websites used:

Desmos, Inc. Desmos Calculator. https://www.desmos.com/calculator

Wolfram Research. Wolfram Alpha. https://www.wolframalpha.com/

EqsQuest, Ltd. Symbolab. https://www.symbolab.com/solver/derivative-calculator