Computing Assignment 4

Daniel Todd

301428609

D100

# What I did:

Text

Description automatically generated To get accurate M for my bisection method, I calculated the first 3 roots manually. I then used the difference between root 3 and root 2 to calculate my interval [a, b] for the 4th root, where a4 = a3 + diff, and b4  =  b3  + diff, and diff is the difference, in terms of x, between p3 and p2. I then automated this process, to generate the generalized statement: an = an-1 + diffn, bn = bn-1 + diffn, where diffn = |pn-1 – pn-2|. This worked for relatively small M, but as I increased the size of M to over 200, this became an issue as the interval would result in same signs for J0(an) and J0(bn). To resolve this issue, I added a small section of code that increments bn by one until the signs for J0(an) and J0(bn) are opposites. I then added some code to check that the error produced by the bisection method did not go over 1x10-5. I also set my tolerance(TOL) for the bisection method to 1x10-12 as this is the lowest tolerance that allows the script to run in reasonable time on my computer. Once this was all in place, I increased the value of M gradually to settle at M = 5000, as this is the maximum M that allows the script to produce results in reasonable time. The error produced by the bisection method also stays below 1x10-5 for this value of M. The segment of code being referred to in this paragraph is below.

I wanted the program to execute efficiently (less than 5 seconds), while also aiming for more accuracy by picking a large enough M and small enough TOL to get accurate results. In this scenario, the more accurate I can make M and TOL, the more robust my script is, as these have a bounding effect on the error for my bisection method.

For the determination of my a and b values, I wanted to be as efficient as possible, which I did by automatically adding the difference between the previous two roots. However, this method of determining the a and b values did not prove to be robust, and as such I added the incrementation algorithm for b previously detailed. This method for finding a and b is also accurate, as I noticed the difference between roots stays relatively the same, only slightly decreasing overtime, and as such I calculated diff dynamically.

# The Results:

I then treated the roots produced by the script as output values for some input m where, m ∈ {1, 2, …, M}, and found the linear least squares interpolating polynomial for all M points. I then used the concept of extrapolation to find α and β from this interpolating polynomial, where α is the slope and β/α is the y-intercept, asking MATLAB to print the calculated values for α and β, I get the results α= 3.14159258220125, and β= -0.249919185192266. With these results I could immediately recognize that the exact value for α is likely to be , and the exact value for β is likely to be -1/4.