MACM 316 – D100
February 11, 2019

Kai Sackville-Hii (301310336)
ksackvil@sfu.ca

Computing Assignment 4

Finding the Roots of Bessel Functions

The goal of this assignment is to compare Matlab's *fzero* function and Newton's method when finding the roots of the Bessel Functions - $J_m(x)$.

Matlab's fzero and Newton's method

For Matlab's *fzero,* each iteration requires an interval $[a_{m,k}, b_{m,k}]$ in which to search for the root. Similarly, Newton's method requires an initial guess $X_{m,k}$, which is close to the desired root. Using known properties of the Bessel Functions, the initial parameters can be chosen intelligently. The first zero of $J_0(x)$ is known to be around 2.5. The initial interval for *fzero* must bracket this route; therefore, $a_{0,0}$ should be slightly smaller than 2.5 and $b_{0,0}$ should be slightly larger than 2.5. For Netwon's method the initial guess should be close to the root. Therefore, let

$$[a_{0,0}, b_{0,0}] = [2.4, 2.6],$$
$$X_{0,0} = 2.5.$$

The kth root of $J_m(x)$ is known to be larger than the previous root by around π. For *fzero* our kth interval must bracket the kth root so let

$$a = k^{th} \text{ root} - \text{offset}$$
$$b = k^{th} \text{ root} + \text{offset},$$

where offset is a value of π/2. For Newton's method the kth root estimate was used as the initial value for the kth interval. The first root of $J_m(x)$ - excluding $J_0(x)$ – is bracketed by the first two roots of $J_{m-1}(x)$. Therefore let, the interval for *fzero* be the first two roots of $J_{m-1}(x)$, and the initial guess for Newton's method be halfway in-between these wo roots.

Comparing the Methods

For each iteration of both methods, the number of function evaluations was recorded. With a tolerance of 1e-6, *fzero* has an average of 6.23 function evaluations, whereas Newton's method has an average of 8.93. Decreasing the tolerance to 1e-10 increases the average function evaluations for both methods; however, *fzero* still has the lesser of the two. From this result, it can be concluded that *fzero* is more efficient than Newton's Method. Experimenting with the initial parameters yielded the following observation: <u>the closer the initial guess/interval is to the actual root, the less evaluations need to be performed</u>. The accuracy for both methods can be measured using the max absolute error given by figure 3000. With a tolerance of 1e-6, *fzero* has an error of 4.8058e-7, whereas Newtons method has an error of 1.2263e-14. Decreasing the tolerance to 1e-10 decreases the error for both methods. <u>Therefore, the smaller the tolerance, the more accurate the root</u>. A robust algorithm should be able to handle bad parameters. For *fzero* a large interval would be a bad initial parameter. With an initial interval of [1, 5] *fero* can find roots but with a substantial loss of accuracy. When the initial interval does not bracket the first root or is large, unexpected results occur. Small changes to the initial guess for Newton's method will cause unexpected results and failures. It can be concluded that <u>*fzero* is more robust than Newton's method, since it can withstand bad parameters to some extent</u>. Figure 1 shows the final frame *CA4_BesselMovie.m*, using *fzero* and smart initial parameters to find the roots.

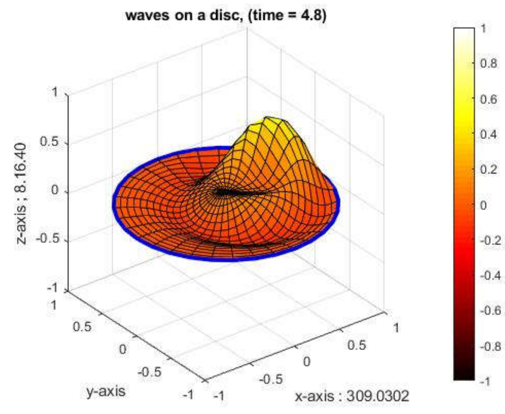

Figure 1: Final Frame from CA4_BesselMovie.m