Roy Chan
chanroyc@sfu.ca
301202770

MACM 316 : Polynomial interpolation and node distribution

Introduction

Barycentric interpolation is the rewritten form of the Lagrange basis polynomial rewritten as the following:

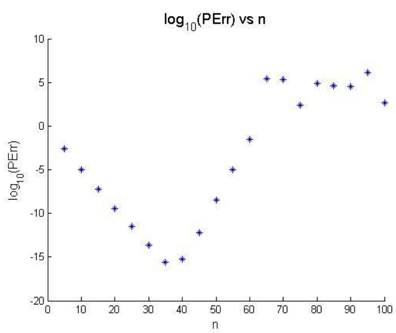
$$\ell_j(x) = \ell(x)\frac{w_j}{x - x_j}$$

The weights w_j could be pre-computed which leaves the running time at $O(n)$ as opposed to $O(n^2)$ for computations with the Lagrange interpolation method.

Discussion

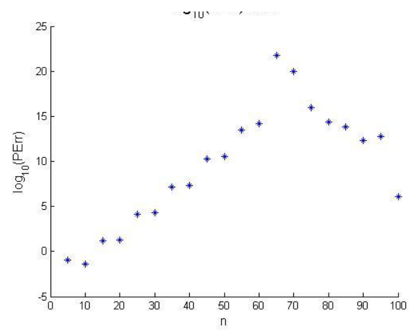
For the code of this report, I have utilized *InterComp.m* from week 8 as the basis of my code. The main source file of this code is *asn6.m*, where *baryinterp.m*, *baryweights.m* and *barycweights.m* are functions utilized in the code.

```
average =

   0.971972018044558
```




```
average =

  -3.582217516026662
```

Figure (1). Function 1




```
average =

   9.435342403601148
```

Figure (2). Function 2

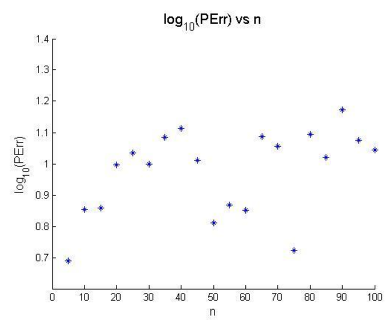


Figure (3). Function 3

The following results are based on possible incorrect code, however, my hypotheses on this report will remain to be true to the results I have acquired from my code on MATLAB. *Figure (1)* , *(2)*, and *(3)* compute the $\log_{10}$(PErr). The averages are given above. Based on the above graphs created from my MATLAB code, the general tendency results in larger errors for larger sizes of *n*. For Function 1, the error dips down before increasing as *n* increases. The accuracy of the equally spaced nodes are more accurate when *n* is small.

In *Figure (3)*, *baryweights.m* is replaced with *barycweights.m* which uses the known formula given for Chebyshev nodes. As we can see from comparing the averages, the accuracy of the interpolations are in the following order from highest to lowest: Function 1, Function 3, and Function 2 (Function two has an order of 2, which may be the reason why the accuracy is quite off). Relative to the other two functions, Function 3 is robust. As a result of incomplete/inaccurate code, the smallest size *n* for which e_n is smaller or equal to 10^{-5} could not be found.