Homework 1 Problem 1

Write a program to compute this sum for a given input N. In your sum, don't compute 2*k* with each iteration. Rather, use a variable which is multiplied by 2 each time through your *N*1 ln(2)=lim∑ *k* summation loop. Why do the computation this way?

By Horner’s Rule, I can reduce computation time by minimizing that number of computations required by an algorithm. If I only multiply by 2 for k loops, I’m doing k computations where’s if I do 2^k for k loops I’m doing k! computations (significantly more than necessary).

The plot of error shows two regimes: a decreasing error for N = [1, 64] and then a flat-line error for N > 64. Please explain what is going on in each regime.

As we increase N, we quickly reduce the size of the error and thereby increase the precision of the approximation. At N > 64 there is no more precision to be made by increasing the number of terms since floats are numerical approximations with gaps along the Real number line. So, 1E-16 is the best we’re going to do.