syasar

CSE 13S

October 25, 2020

ASGN2 WRITE UP

My Sin and Cos functions were accurate as an approximation, having no difference for the majority of its execution in the range [-2π, 2π). Deltas from the math.h library were only seen from outward spread in the middle of ranges [-2π, 0) and (0, 2π). This is mostly likely due to the normalization of x values where if x was less than -π or greater than π, I would add and subtract 2π, respectively. Otherwise, my Sin and Cos approximations were largely accurate in part to setting them as double variable types, using the less intensive horner form with 14 terms, and setting x^2 as a variable to avoid calling the power function more times than needed and introducing possible error.

My Tan functions were completely accurate when compared to the math.h library. No deltas were seen in range [-π/3, π/3). This most likely due to using a horner form with 14 terms, using a double variable type, and setting x^2 as a variable to avoid extra, unnecessary instructions. Although x was normalized to be in [-π/3, π/3), the functions accuracy was not a result of that at the test was already in the same range.

My Exp function also showed high levels of accuracy with no error greater than 10e-8. This accuracy is most likely due to using a for loop that conducts a summation of e^x, adding each term to make it closer to the math.h exponential function. Once the next term is less than 10e-9 and is unnecessary to add to the summation as it is so small, the function exits the for loop and returns the summation. Its accuracy is also boosted thanks to using long doubles.