Assignment 3

WRITEUP: A Small Numerical Library

Linhao Chen
CSE 13S - Fall 2019

Date:10/20/2019

1 Sin(x) and Cos(x)

Padé approximant is also using some terms of Taylor series to estimate the value of sin(x) and cos(x). The curious thing is the difference between the value I got and value from math library will be bigger when x close to 2pi and -2pi. I think the best reason to explain this is because the absolute value of x is bigger, so the deviation of approximation will be bigger. This situation is really like the thing when we calculate exponent. If we only use 10 terms to estimate the value, it should be fine if x smaller than 2, but it is not enough to estimate the number x like 10 or more. So, if we can use more terms to estimate the value of sin(x) and cos(x), I believe the difference when using big value of x will be eliminated.

2 Tan(x)

The value of tan(x) is almost same. We use sin(x) / cos(x) to get the value of tan(x). The domain of tan(x) is from -(pi/2-0.001) to pi/2-0.001. In this range, the value of sin(x) and cos(x) are almost same with library. That is why the value I got is really similar to the answer from math library in C.

3 e^x

In DESIGN document, I try to use 30 terms to estimate the value of e^x. However, in the real test, when we use the 10-digit decimal precision, we can easily see there still has a huge difference (even the difference is lower than 0.001). So, in the program, I change the 30 terms to 45 terms for eliminating. For 0.0 to 10.0, by using 45 terms to estimate the value, it seems the different between my library and C<math.h> is smaller. I guess the library in C may uses more than 100 terms to calculate the value of exp(x) since I notice the precision of the library is really more precise than what I use especially when I 30 terms or less. If x become larger than 10, we should use more terms to estimate the value since the precision of 45 terms is not enough.

But I still curious about the difference. Although the values are close, sometimes the difference is 0.0000000000. In my opinion, I think it always should be -0.0000000000 because I just uses 45 terms, the precision is worse than math library in C. So, I may miss

something after 45 terms, the real value should be a little bit bigger than what I got. (I think 0.0000000000 means the estimate value is bigger than the value of math library, vice versa) But sometimes the difference is 0.0000000000 and sometimes it will be -0.0000000000. I think it should be problem.