For sin, cos, tan, and exp my differences were all zero. Therefore my value calculated and the value from the library matched up for every single x value inputted into the sin, cos, tan and exp.

Originally I used pade approximation for Sin and Cos but they were inaccurate and their differences were less than .1 because The order 10 pade approximant starts to deviate a little bit around when value of x equals 6. The same case is true for Cos

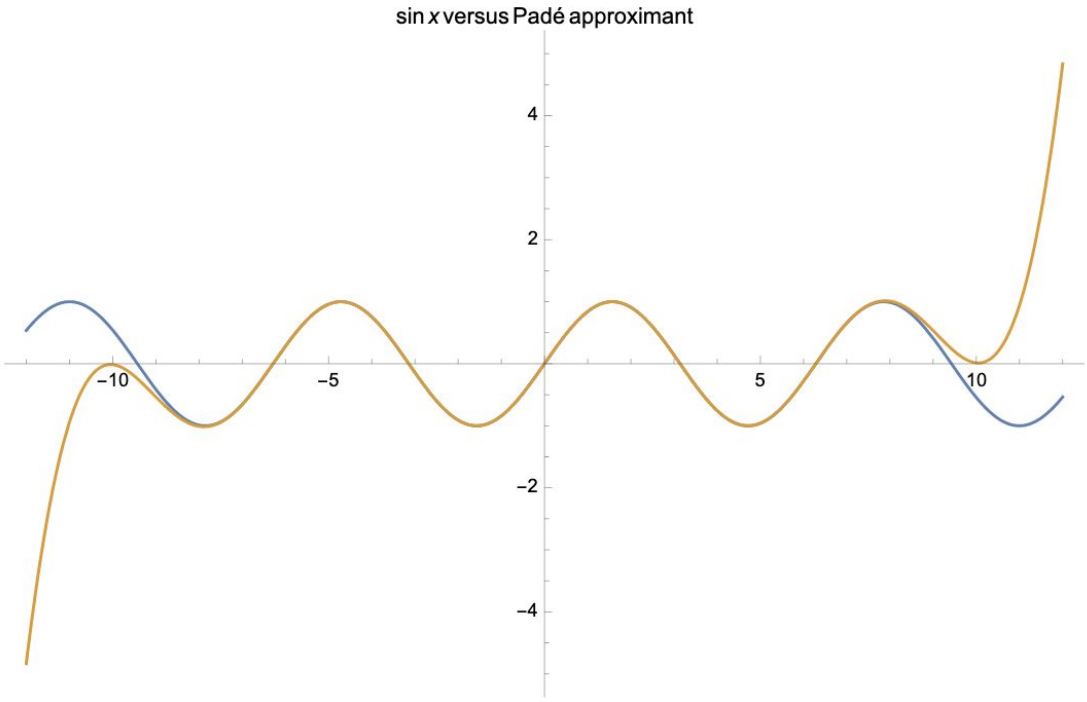
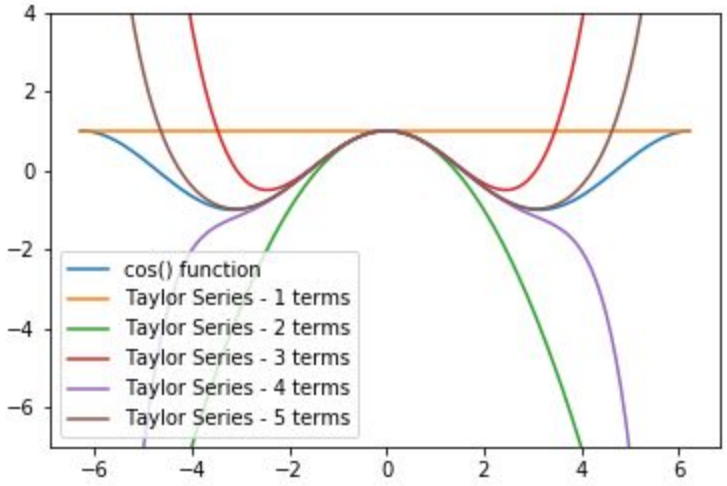


Figure 3: Comparing $\sin(x)$ with an order 10 Padé approximant.

To refine my cose, I used taylor series approximation for sin, cos, exp where summation of 0 to 100 was used. The values would get more accurate as the upper approximation bound would be increased because it sums more terms to makes the summation more accurate.. Initially, I used 14 as the upper bound but it wasn't 100% accurate therefore I changed the upper to be 100.

I used pade approximation given in the assignment for the tan function value. The values were accurate, therefore I did not need refinements in my code.



This shows that the taylor series gets inaccurate as the number of terms decrease.
Therefore I used 100 terms in order to calculate the accurate value for sin, cos and exp.

This also shows that 10 terms is not enough for the summation of sin, cos, and exp because it would deviate from the curve that we want. Therefore more terms the more accurate it is.

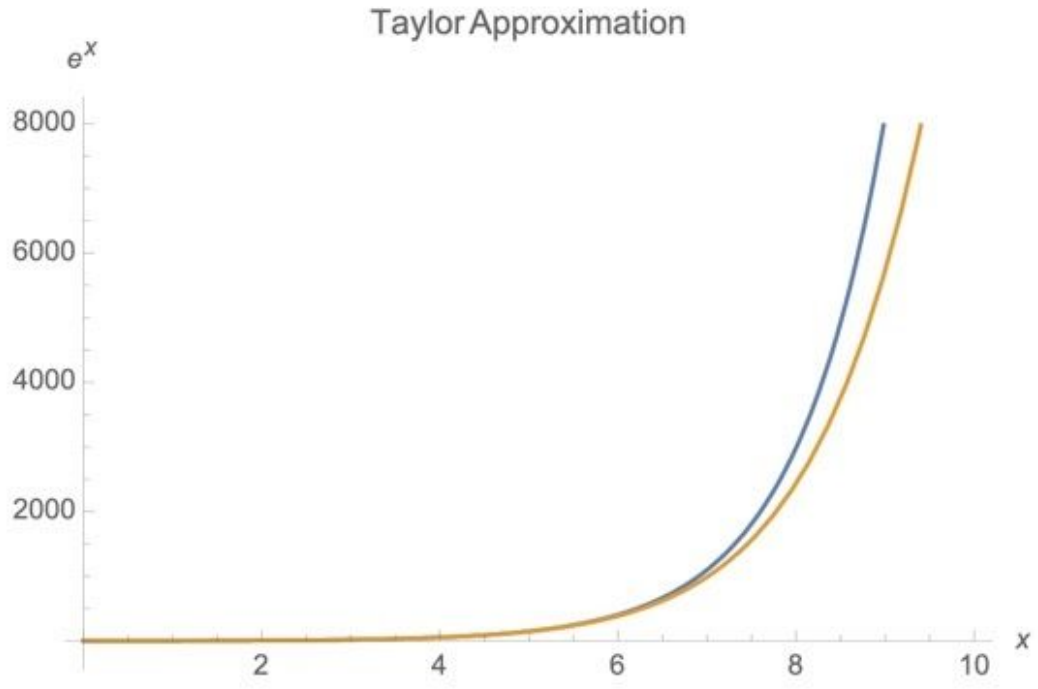


Figure 4: Comparing e^x with its Taylor approximation centered at zero.

Ultimately, my program doesn't have any differences in subtracting the computed value from value from math.h library for sin, cos, tan, and exp.