Nicolas Ayala
nmayala@ucsc.edu

Chart A:

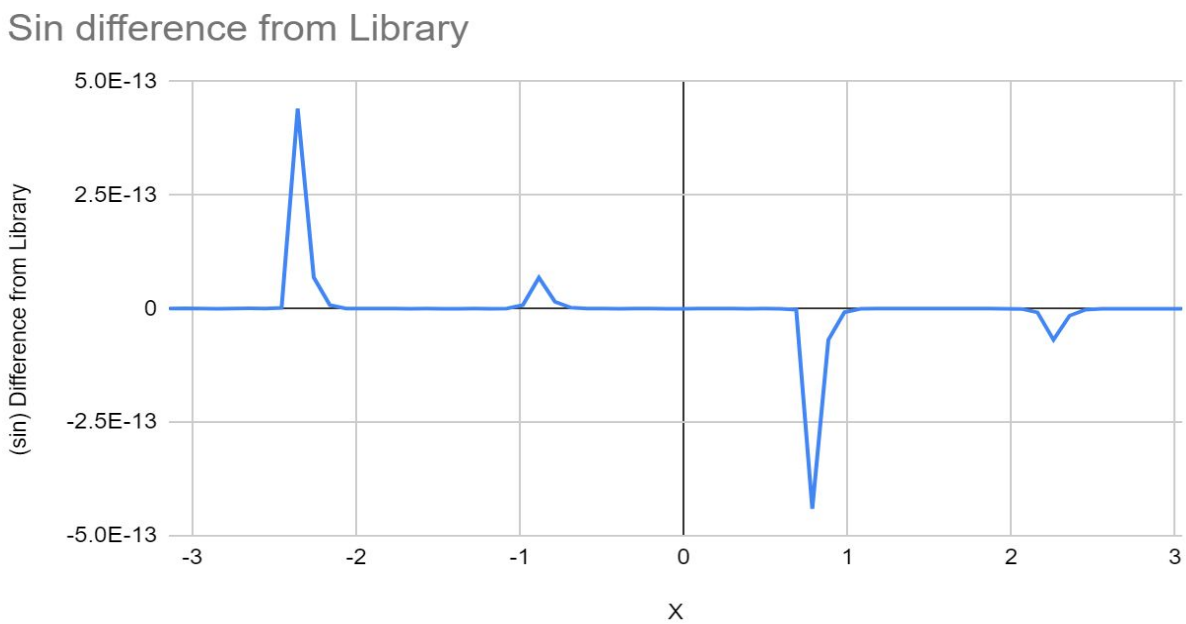


Chart B:

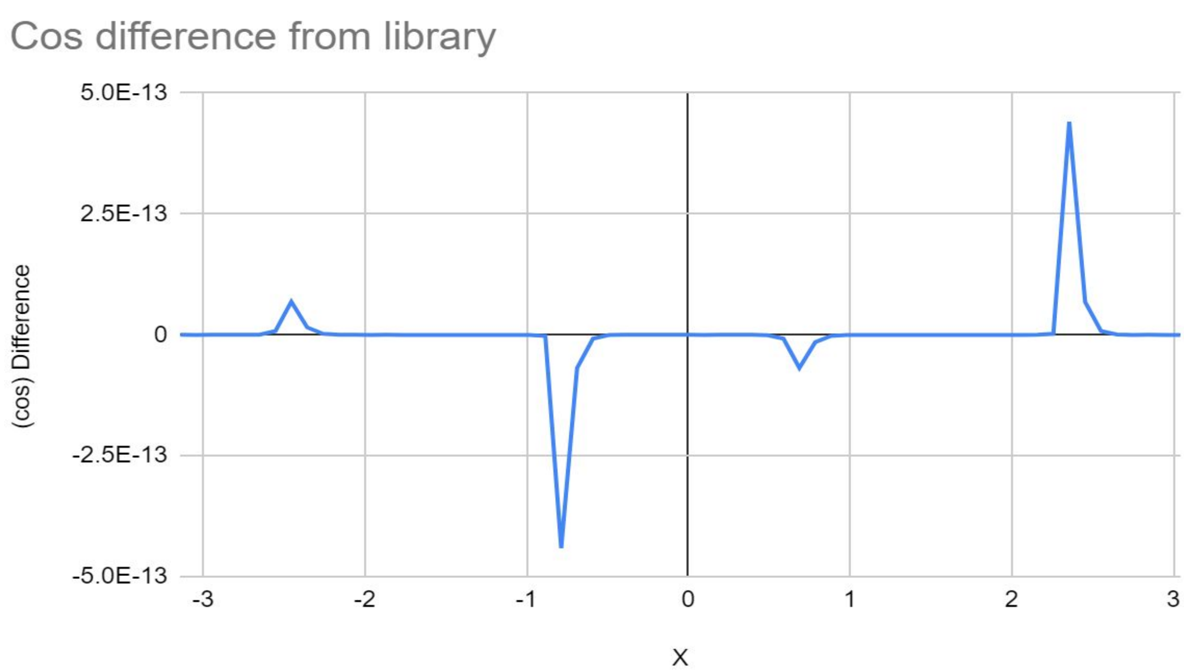


Chart C:

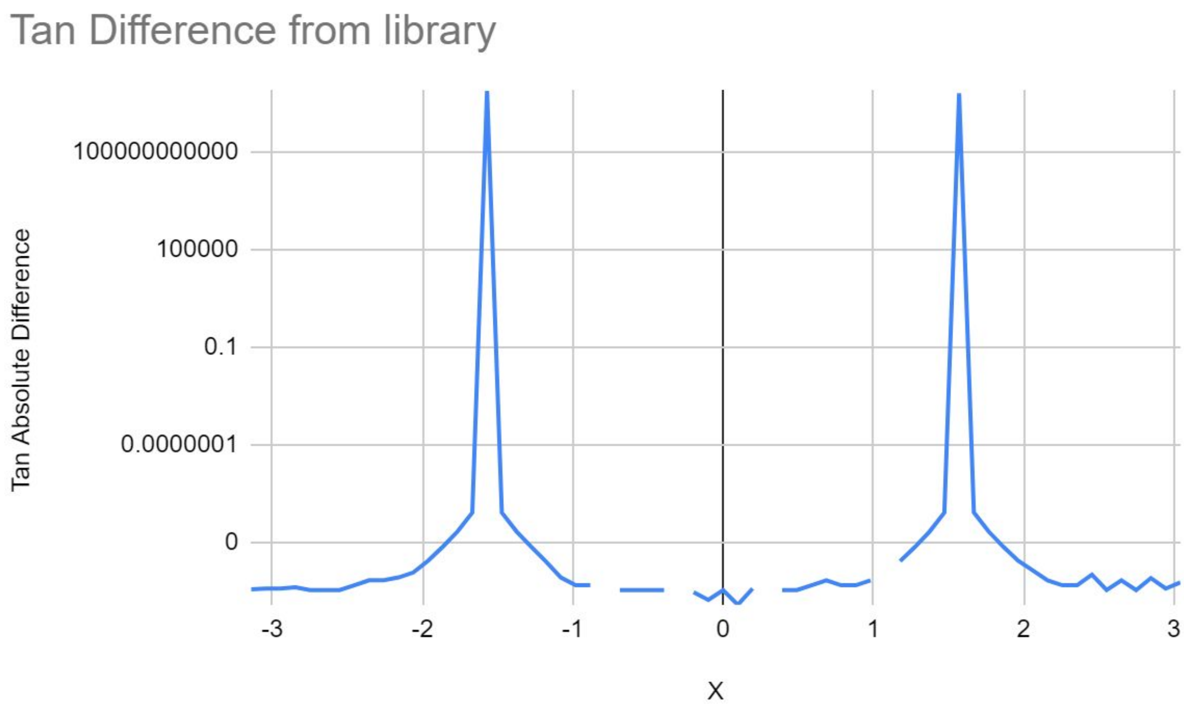
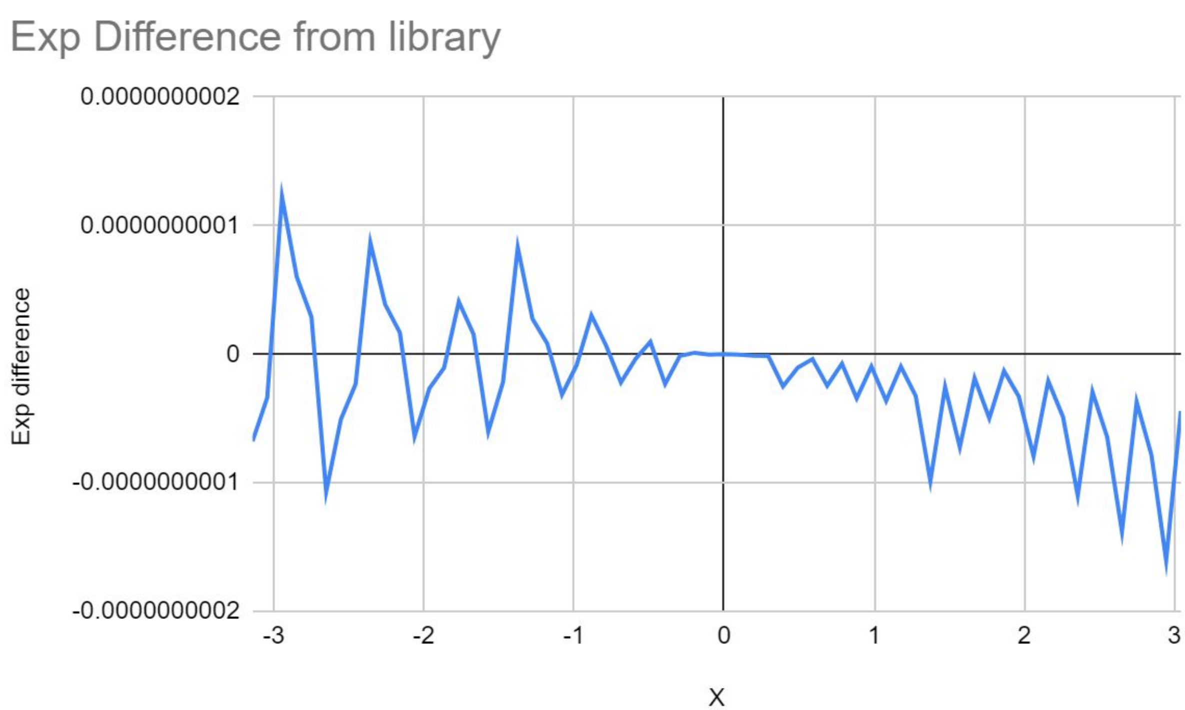


Chart D:



Description

Chart A shows the difference between my implementation of sin, and the implementation of sin found in math.c within the range $(-\pi,\pi)$.

Chart B shows the difference between my implementation of cos, and the implementation of cos found in math.c within the range $(-\pi,\pi)$.

Chart C shows the absolute value of the difference between my implementation of tan, and the implementation of tan found in math.c within the range $(-\pi,\pi)$. Displayed on a logarithmic scale.

Chart D shows the difference between my implementation of exp, and the implementation of exp found in math.c within the range $(-\pi,\pi)$.

Analysis

As you can see in chart A, there are massive(comparatively) spikes around $\pi/4 + k\pi/2$, where k is an integer. My sin was implemented in such a way so that the largest absolute value passed into a Padé approximation was $\pi/4$. It choose between both the sin and cosine padé so as to be most accurate. When, for example, $\pi/4$ was passed into sin, it would pass the value of $-\pi/4$ into the padé of cosine. This is why the results get inaccurate around $\pi/4 + k\pi/2$, as these would pass an absolute value of $\pi/4$ into a padé. Padé approximations are constructed to be more accurate the closer to 0 the input is. As the input strays from 0, the approximation becomes increasingly inaccurate, because the error term grows very fast with respect to the input.

My implementation of cosine simply translates the input and then passes it into sin. You can see that chart B is a translation of chart A.

I did nothing fancy for my tangent implementation. You can see in chart C that when $\pi/2$ is passed into it, the approximation becomes wildly inaccurate. So inaccurate that I had to use a very large logarithmic scale to graph it. This inaccuracy is because the real tangent function has an infinite discontinuity at $\pi/2 + k\pi$ where k is an integer, yet the padé approximation does not.

My implementation of exp was also nothing fancy. I do not know for sure why the line appears jagged. My guess is that the line is jagged because the approximation is oscillating between having a term that is just barely under epsilon, and having a term that is well below epsilon.