FIT3139 Assignment 1 Q 1 A

By Ian Tongs

May 7, 2020

Performance over an integer domain:

It should be noted that, for relative absolute error (%), I have followed the formula as given in the lecture notes, so negative values are possible.

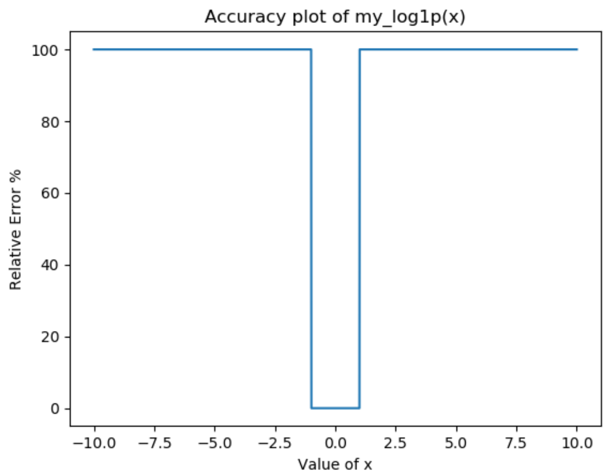

Fig 1a.1: relative absolute error over a wide domain

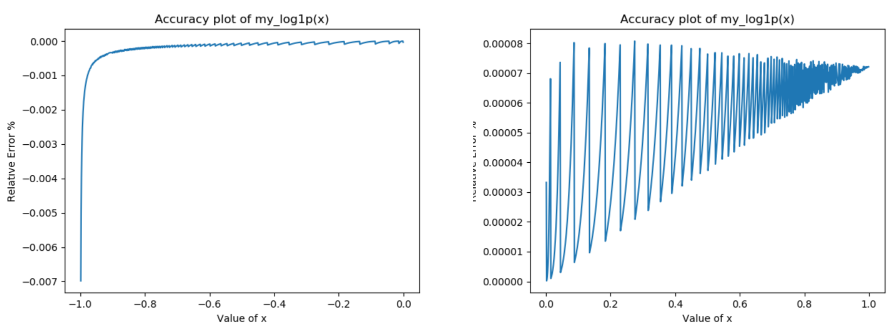

Fig 1.a.2: relative error between -1 and 0 (left) and 0 and 1 (right)

From the plot, we can clearly see that $my\_log1p(x)$ is most effective between the values of -1 and 1 inclusive. Upon further investigation, input values outside this range fail to converge, oscillating further and further from their actual values. Thus, for operations where we only wish to find the outputs between -1 and 1, $my\_log1p(x)$ is brilliant, but outside this domain it is pretty useless. It is worth noting that positive values (on the domain $[0, 1]$) are more accurate than those on the domain $[-1, 0]$ by up to a factor of around 10^2.

Compared to the in-built $log(1 + x)$, upon further investigation, $my\_log1p(x)$ converges to the correct value quickly within its accurate domain of $[-1, 1]$, which would give it an advantage over $log(1+x)$ in this range as this method would be computationally less expensive than a method that could handle a wider domain. However, outside this range, the in-built function would be clearly superior.

Performance over a machine epsilon domain:

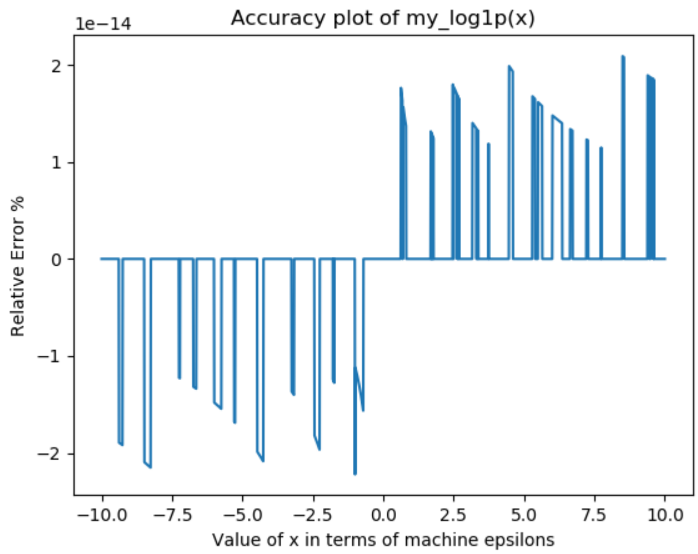

Fig 1a.3: relative absolute error over a machine epsilon domain

On a domain on terms based on machine epsilon, we can see that $my\_log1p(x)$ handles inputs on the level of or below machine epsilon very well, as relative error percentages are on the level of $10^{-14}\%$, which indicates the relative error is exceptionally small. In non-percentage form, this puts the error itself around or below machine epsilon itself.