FIT3139 Assignment 1 Q 1 B

By Ian Tongs

May 7, 2020

Expansion of $log(1+x)$:

Firstly, we must find an expression for y in terms of x:

$$\begin{aligned} 1+x &= \frac{1+y}{1-y} \\ 1-y+x-x\cdot y &= 1+y \\ y+y+x\cdot y &= x \\ y &= \frac{x}{x+2} \end{aligned}$$

We can also expand our log expression as given in the question:

$$\begin{aligned} log(1+x) &= log\left(\frac{1+y}{1-y}\right) \\ &= log(1+y) - log(1-y) \\ \therefore log(1+x) &= log(1+y) - log(1+(-y)) \end{aligned}$$

Combining this with the Taylor Series expansion given in part A, we can say:

$$log(1+x) = \frac{x}{x+2} - \frac{-x}{x+2} - \left(\frac{1}{2}\cdot\left(\frac{x}{x+2}\right)^2 - \frac{1}{2}\cdot\left(\frac{-x}{x+2}\right)^2\right) + \cdots + \frac{(-1)^{n+1}}{n}\cdot\left(\left(\frac{x}{x+2}\right)^n - \left(\frac{-x}{x+2}\right)^n\right)$$

Which gives us the formula to use in $my\_another\_log1x()$.

Performance over an integer domain:

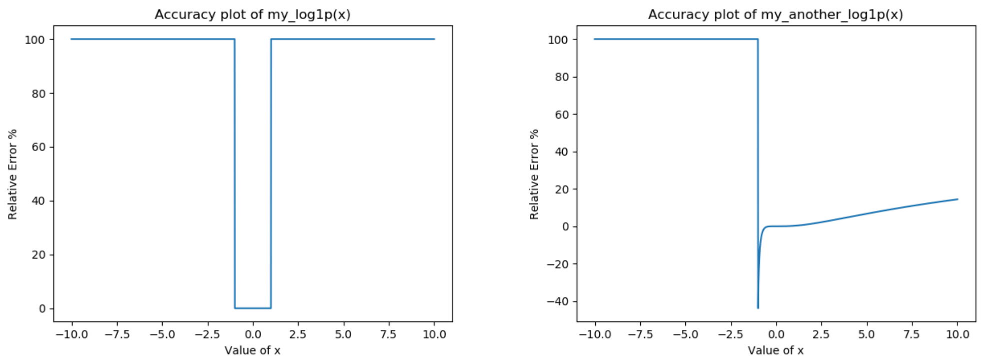


Fig 1b.1: relative absolute error over a wide domain, for $my\_log1x()$ on the left and $my\_another\_log1x()$ on the right

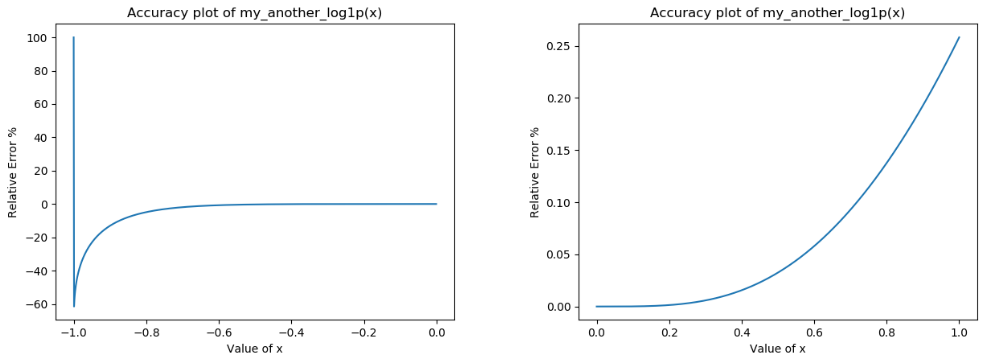
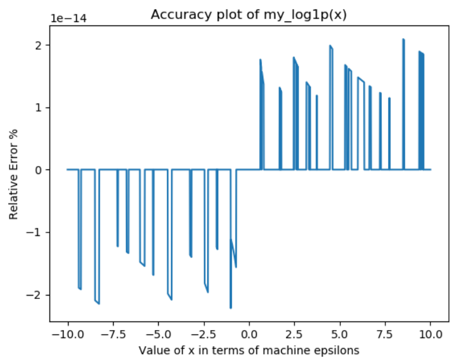


Fig 1b.2: relative error between -1 and 0 (left) and 0 and 1 (right), for $my\_another\_log1x()$ specifically for both

As we can see from the plots and recalling those from part A, we may observe significantly different behaviour between the two functions. It should be immediately apparent that $my\_another\_log1x()$ has a wider domain over which it converges than $my\_log1x()$, however the relative percentage error quickly grows as you move away from zero, to values many orders of magnitude greater than what is given by $my\_log1x()$ within the domain $[-1, 1]$. This is especially prominent near -1 where $my\_another\_log1x()$ starts with a relative percentage error around -60%, which is far 'greater' than what $my\_log1x()$ would produce. Thus, for values in the domain $[-1, 1]$, $my\_log1x()$ has superior accuracy.

It must be noted however that $my\_another\_log1x()$ converges for values greater than 1 and with mostly low error error when not far above 1. Thus, if dealing with x values in a domain like $[0, 2]$ (or really any x value greater than 1), $my\_another\_log1x()$ produces more accurate results.

Performance over a machine epsilon domain:



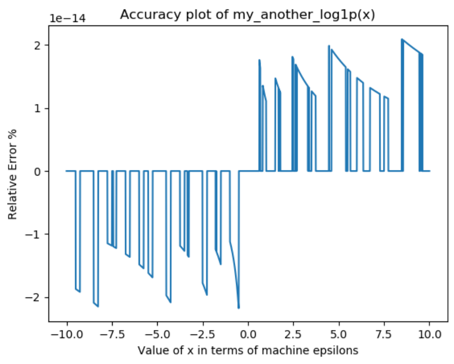


Fig 1b.3: relative absolute error over a machine epsilon domain, for $my\_log1x()$ on the left and $my\_another\_log1x()$ on the right

From the plots in the order of magnitude of machine epsilon, we can see that $my\_another\_log1x()$ and $my\_log1x()$ behave similarly on this domain of inputs. It can be observed that $my\_log1x()$ appears to perform better than $my\_another\_log1x()$, appearing to have fewer values where its error is distinguishable from 0 on this range, but at this point we would be splitting hairs.