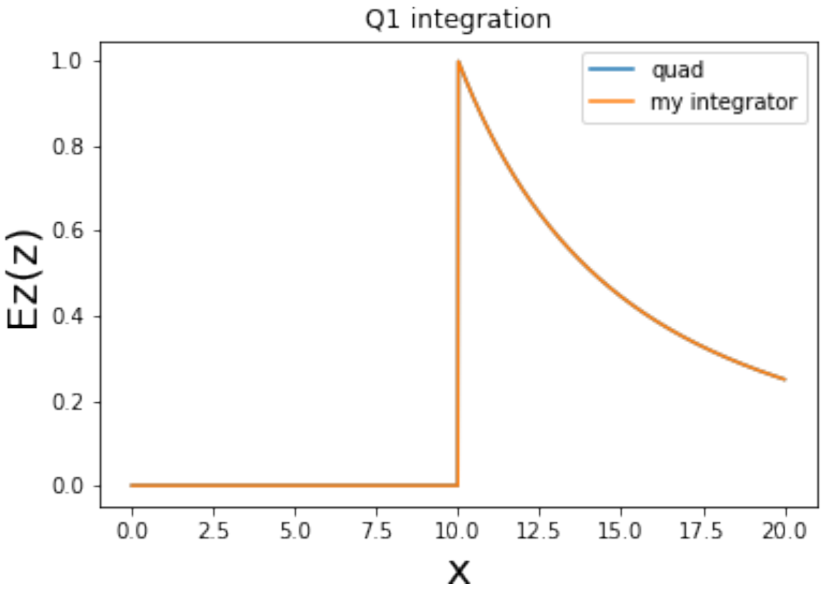
Question # 1



This plot includes the integral for both quad and my integrator. For this plot, I do not have z=R in my x array. If I include z=R, than the kernel of my notebook crashes when using my integrator but works perfectly fine with quad. I set R = 10 and all the constants to 1.

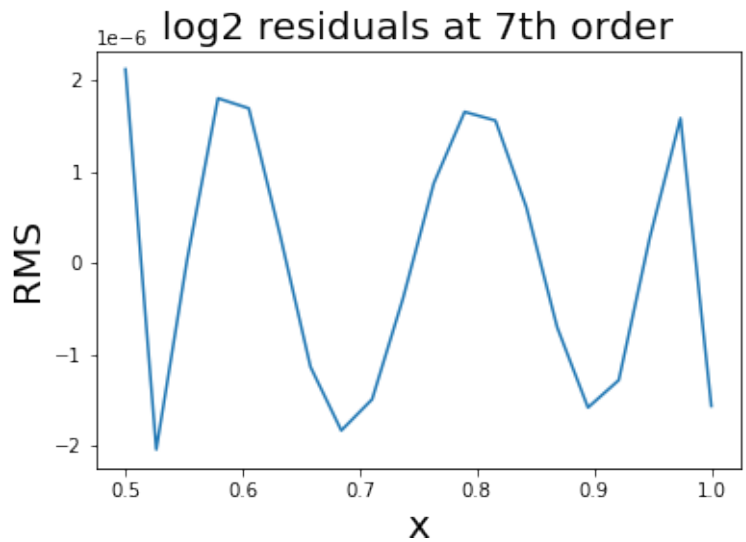
Question 2

For this I don't have much to show. The code for my integrator is a the very beginning of my notebook, because I used it for Q1, it much faster than the one from class. In the value *extra* is an array that stores f(a), f(b), the past simpson evaluation, midpoint, and f(midpoint). This array gets updated every recursion and the value used in the next recursion.

Every recursion I save 3 function calls and one simpson approx. So say the equivalent of 4 function call. So for 10 recursion I save 40 function call.

Question 3

For this problem, I computed the fit using the code from Prof. Sievers. I fed the Cheb fit with an array from -1 to 1 and fed the log function with an array from 0.5 to 1. Here I plotted the residuals from the fit and np.log2(). I show the results using 7th order and 8th order. We see that to get under 10^-6 max error we need to go to 8th order.



Now to compute make the function mylog2 I used the fact that

$x = M \cdot E^2$ where M = mantissa , E = exponents

$$\log_2(x) = \log_2(M \cdot E^2)$$

$$= \log_2(M) + \log_2(E^2)$$

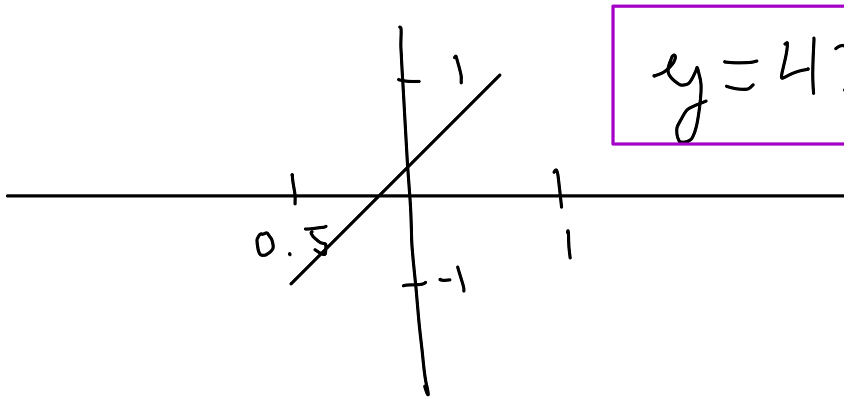
$$= \log_2(M) + E$$

also $\ln(a) = \dfrac{\log_2(a)}{\log_2(e)} = \dfrac{\log_2(M) + E}{\log_2(e)}$

In my code I use the value of $\log_2(e)$ as a constant to save on computation time

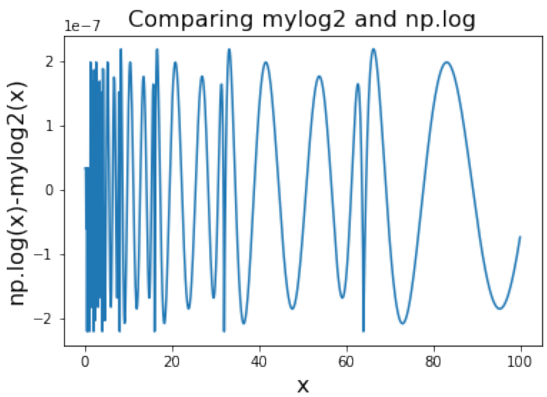
Also I use the following mapping

$(0.5, 1) \to (-1, 1)$



$\boxed{y = 4x - 3}$

Here I plot the
from 0.1 to 100.



residuals from mlog2() vs np.log()