Lab 11A

The initial function that was used for the program was:

The function with no root is:

There is an error with no output from the program due to the program is stuck in a loop of calculations.

## test for x = 3 resulted to no output##

The function with more complex is:

We added by replacing the -1 with . There is only one root that converges to .569

testing with 32 the number of outputs is 16 numbers until reached to a root

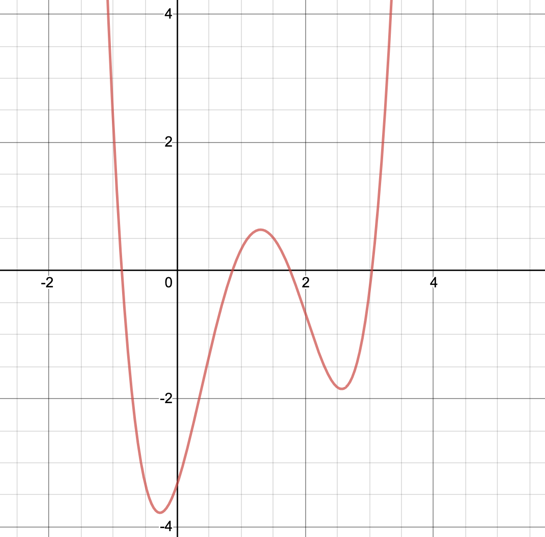
## x = +32, x = -32 ##

“”[32.0, 21.4661717, 14.4379603, 9.7441771, 6.6029311, 4.491604, 3.0604406, 2.0760239, 1.3873138, 0.9141049,0.6498738, 0.5765821, 0.5701476, 0.5698533, 0.5698408, 0.5698403]””

testing with -32 the number of inputs is total of 16 numbers until it reached to a root.

“”[-32.0, -21.1772235, -13.9560693, -9.1328391, -5.9032702, -3.7282089, -2.2430951, -1.1957289,-0.402385, 0.2573456, 0.5771725, 0.5701767, 0.5698546, 0.5698409, 0.5698403]””

both test guess of 32 and -32 does converge into the single root. (Bisection takes 16 times, Newton’s method takes 15 times)

For more complex function y = (5/6)\*x^{4}-4x^{3}+(23/6)x^{2}+3x-(10/3)

Guess: the newton’s method would converge root faster.

For bisection approach:

a, b = [2,4] //(10-6)

result = 3.234836

it takes 21 times to get the result.

For Newton’s method:

Result = 3.234837

It takes 5 times to get the result

So Newton’s method converges to the root faster. But there are some disadvantages of Newton’s method, when we use the Newton’s method, the function that we used has to be continue and the first derivative function exist. And for some points, newton’s method cannot be used.