**Q2):**

a):

For function g1(x), the solution will not converge since |g1’(2)| = 4/3 > 1.

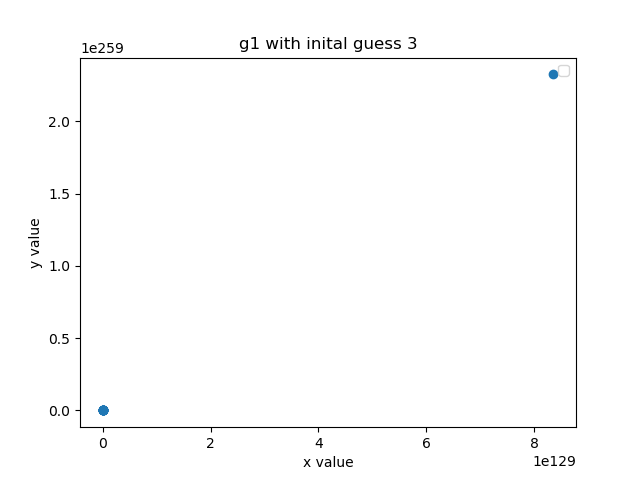
For function g2(x), the solution will converge since |g2’(2)| = 0.75 < 1.

For function g3(x), the solution will converge since |g3’(2)| = 0.5 < 1.

For function g4(x), the solution will converge since|g4’(2)| = 0 < 1 .

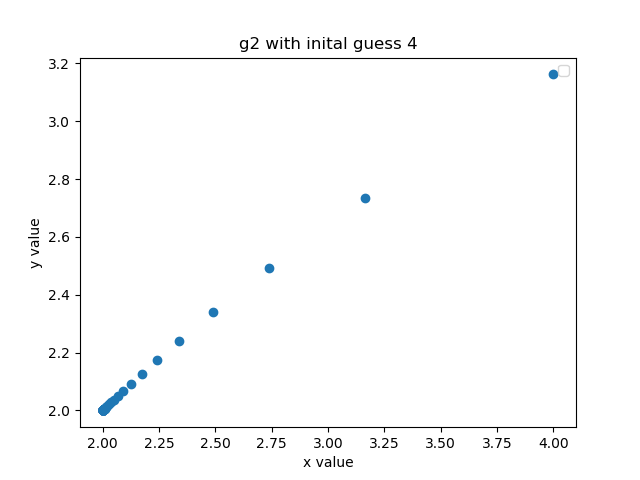
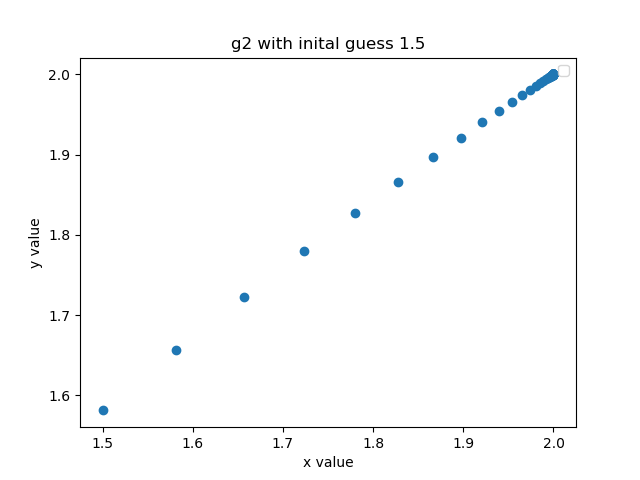
b):

For g1(x), the results of numerical experiment are shown below:



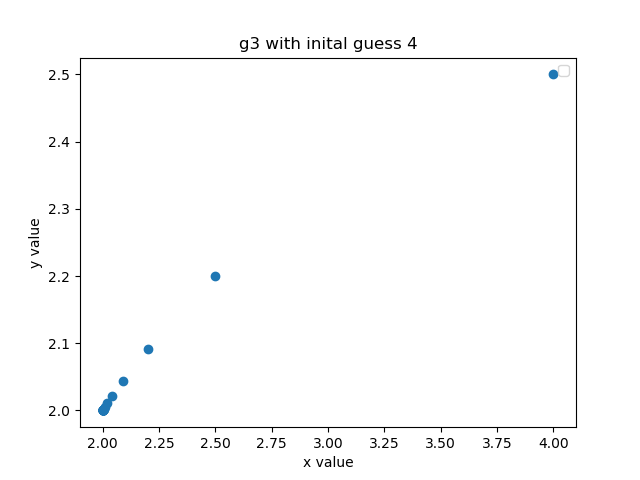
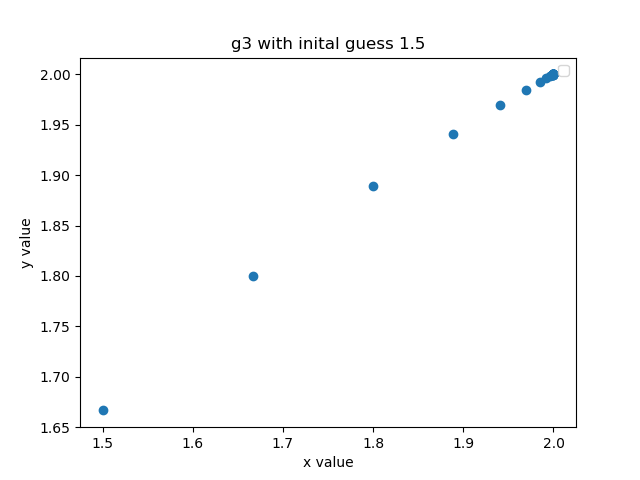
From the graph, we confirms that our solution will not converge to x = 2 since its derivative that that point is greater than 1.

For g2(x), the results of numerical experiment are shown below:



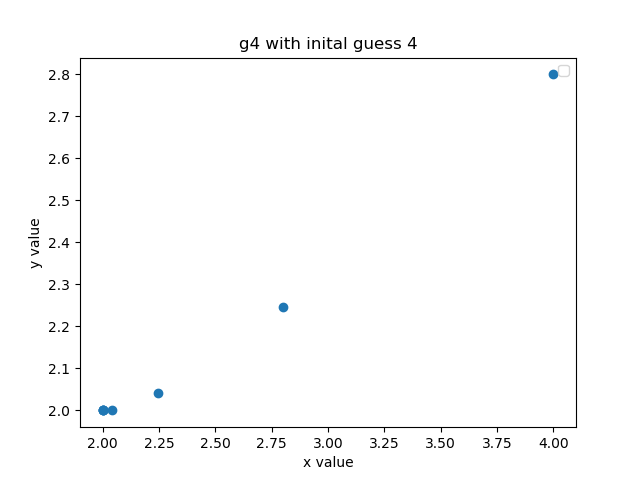
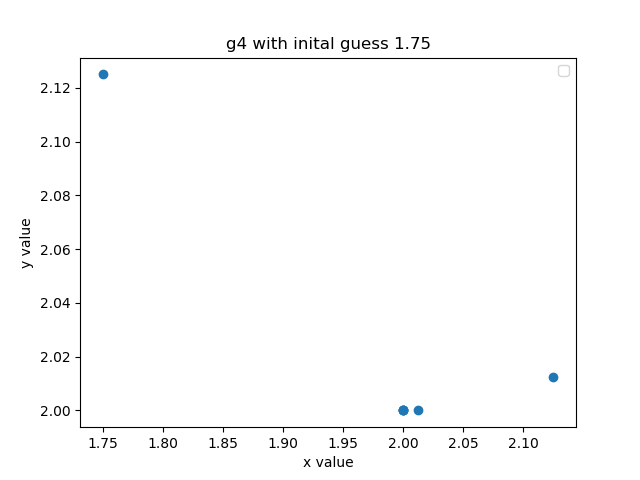
From the graph, we can obersve that the convergence for g2(x) is monoton convergence, which agrees with what we get in part a). From the code in Question 1, the convergent rate and asymptotic error constant κ (1.7095112913514545 and 23887959401.951847) indicate that this might be an superlinear at rate 1.7.

For g3(x), the results of numerical experiment are shown below:



From the graph, we can oberve that the convergence is also a monoton convergence, which agrees with our analytics in part a). The result from our algorithm in Question 1 suggest that this is a linear convergence (r = 1.0 and κ = 0.5).

For g4(x), the results of numerical experiment are shown below:



From the graph we can oberve that g4(x), we obtain a quadratic convergence. The algorithm we create in Question 1 also agree with what we observed. (r=1.9999306, κ = 0.99939059).

c):

Since newtons methed calls for :

Then, can be written as :

Which equals to g4(x) as desire.

**Q3):**

a):

Since , we can easily calculate that its derivative

When we apply the alternative update we proposed, we get:

Since we know the value of m, we can assign . Then, we can calculate the root directly in one iteration.

b):



The above algorithm will run the iteration given by the user and then pass list of x values into the algorithm in Question 1 to obtain an estimation for c and then use it to calculate m. Note that the iteration must be greater than 3 for estimate\_convergence() to produce a result. More iteration means that estimate\_convergence() will return results that more accurate.