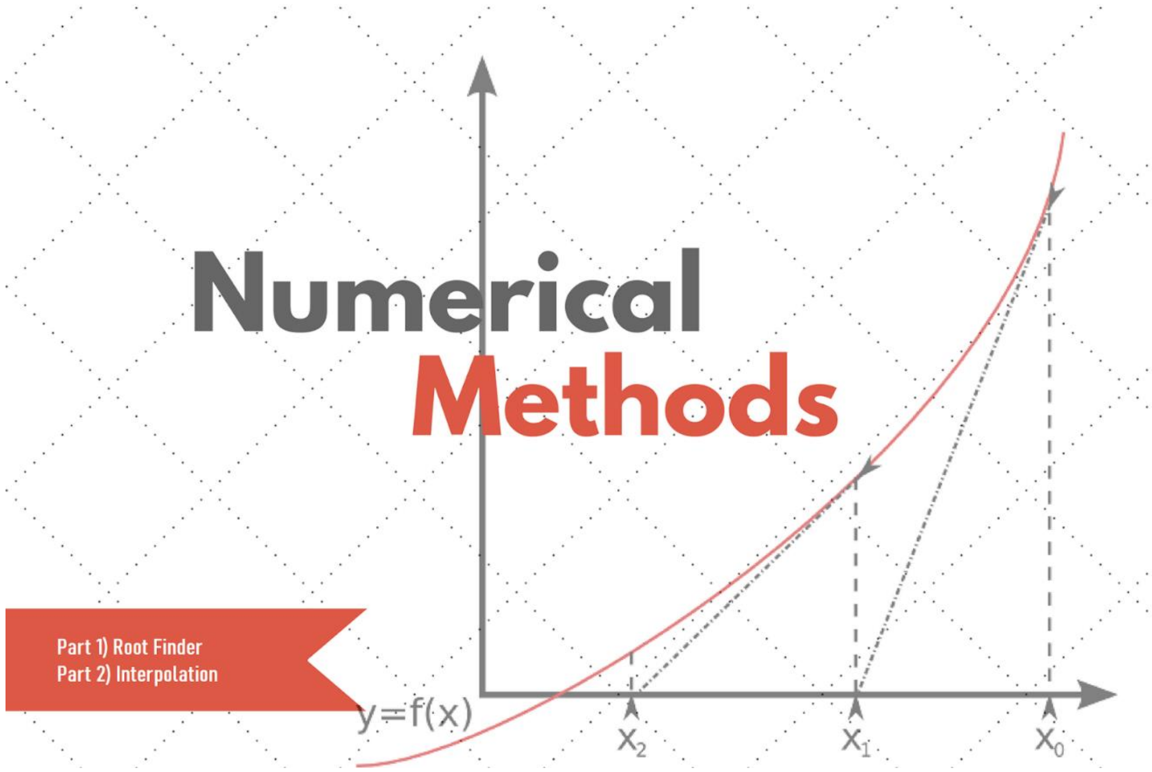
Numerical Methods

Python program to find a root of a function in many different ways



Ahmed Ali Elsayed Saber ID : 07
Ahmed Ayman Mohamed ID: 03
Islam Yousry Abdelwhaid ID: 14
Mahmoud Ibrahim elsayed ID:58
Hamza Hassan Mohammed Ali ID:26

1. Root finder

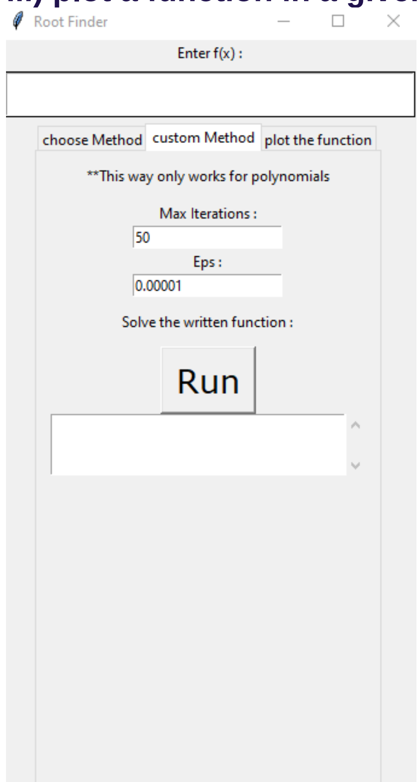
a. General description

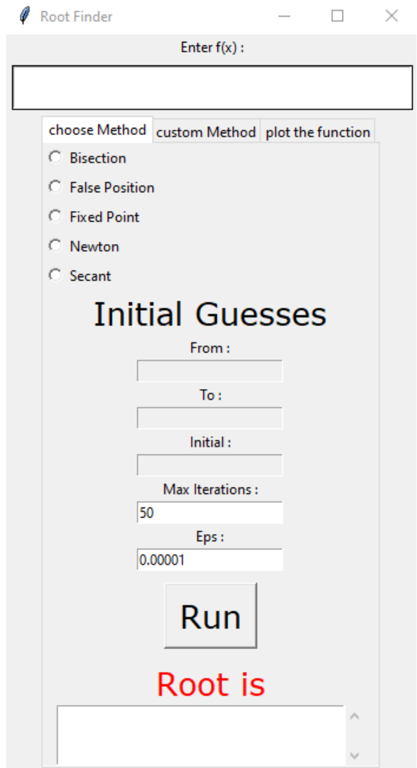
It’s a python program that is divided into 3 parts

i) find a root with a specific numerical method and boundaries.

ii) find all possible roots for a polynomial function.

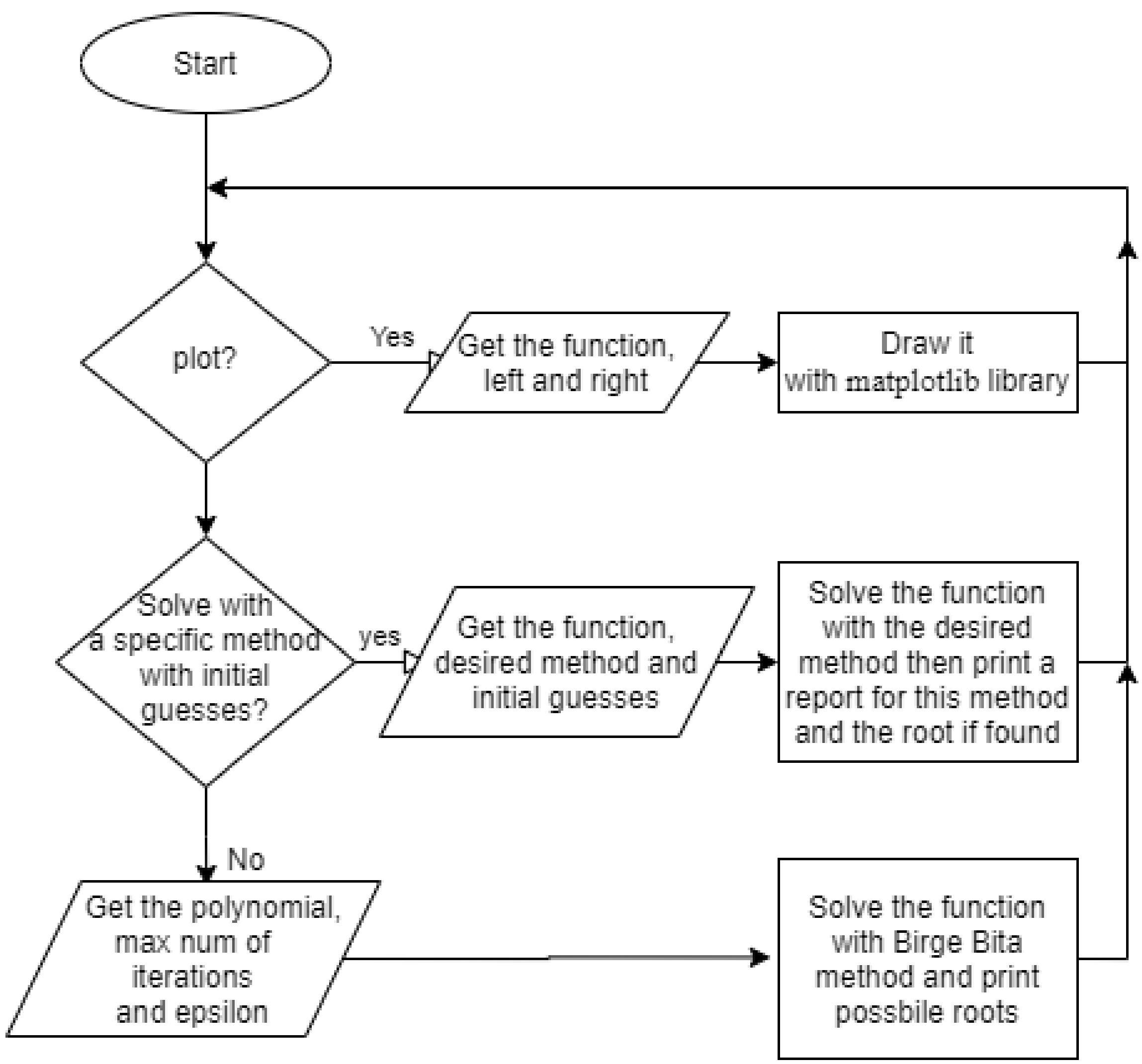
iii) plot a function in a given interval.




Root Finder
Enter f(x) :
choose Method
custom Method
plot the function
Bisection
False Position
Fixed Point
Newton
Secant
Initial Guesses
From :
To :
Initial :
Max Iterations :
50
Eps :
0.00001
Run
Root is

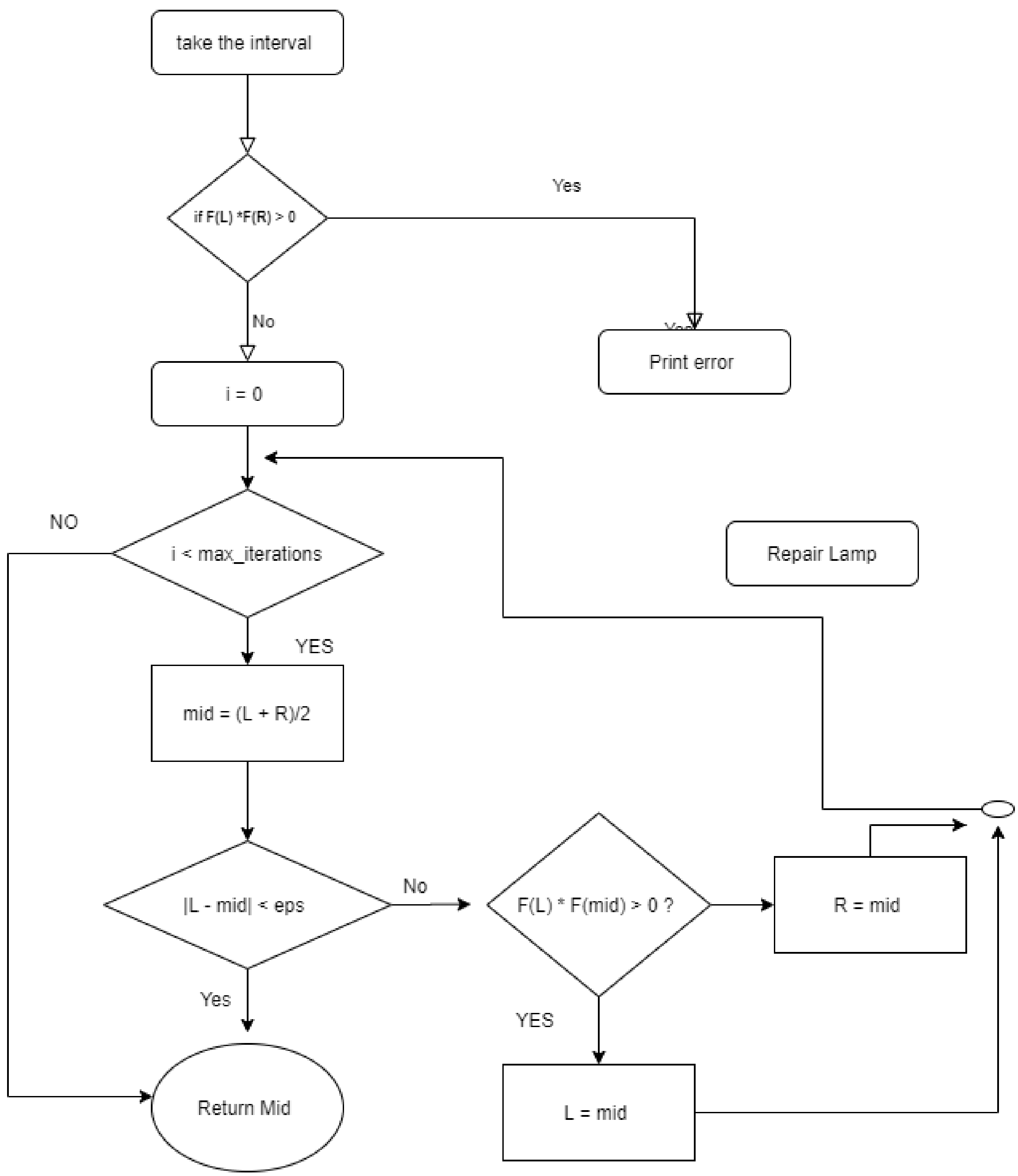

Root Finder
Enter f(x) :
choose Method
custom Method
plot the function
From :
To :
number of plot points :
50
Plot

As it appears in the previous figure.
How does this program work?
It follows the next flow chart.
So at first the user should determine his wish solving any given function that may contain constants, polynomials, power function, logarithmic function, sin, cos , tan…etc.
Or solve a polynomial with Birge Vieta method with no given initial guess. Or plotting the function in a given interval.

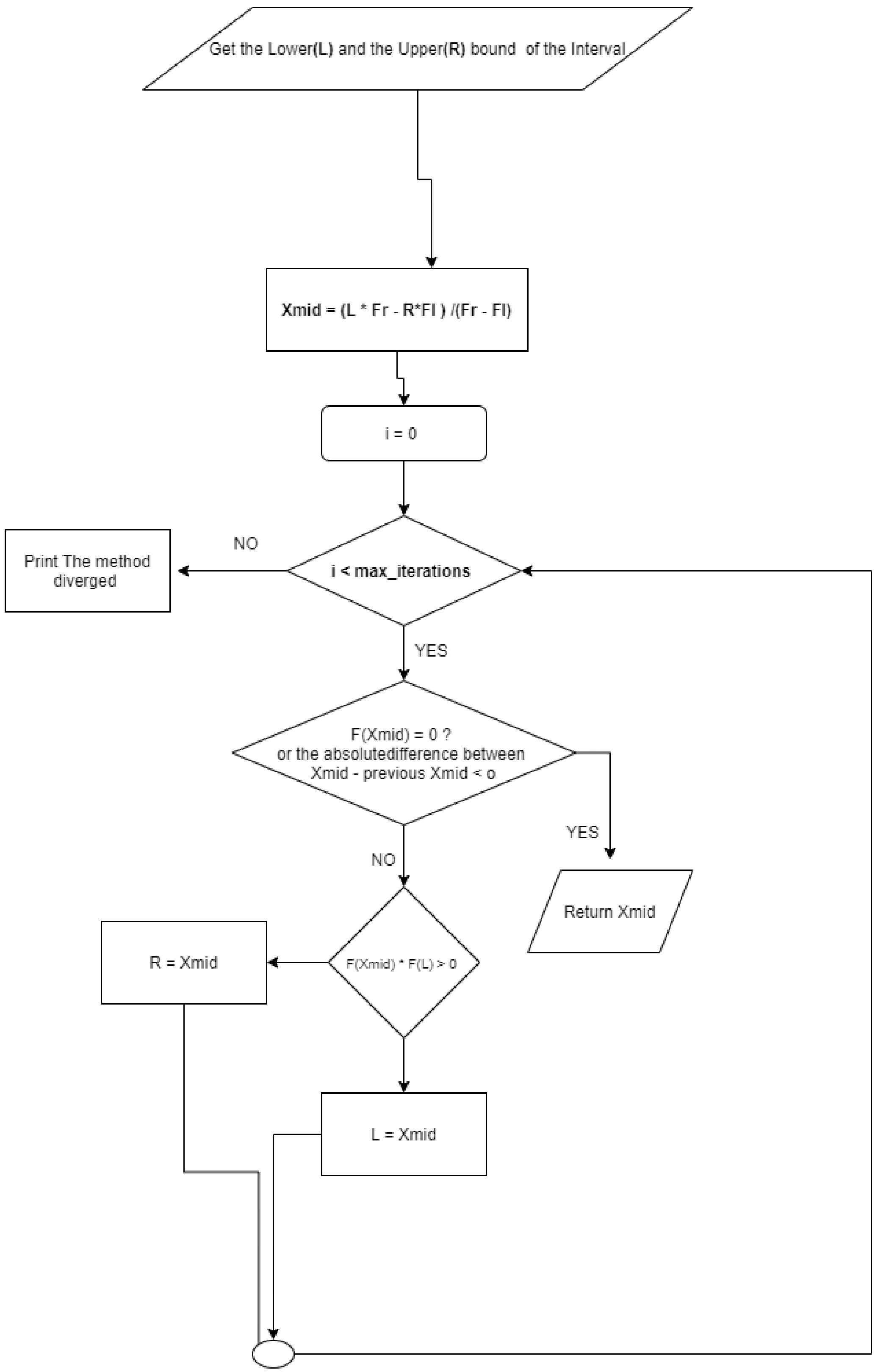


b. Methods description

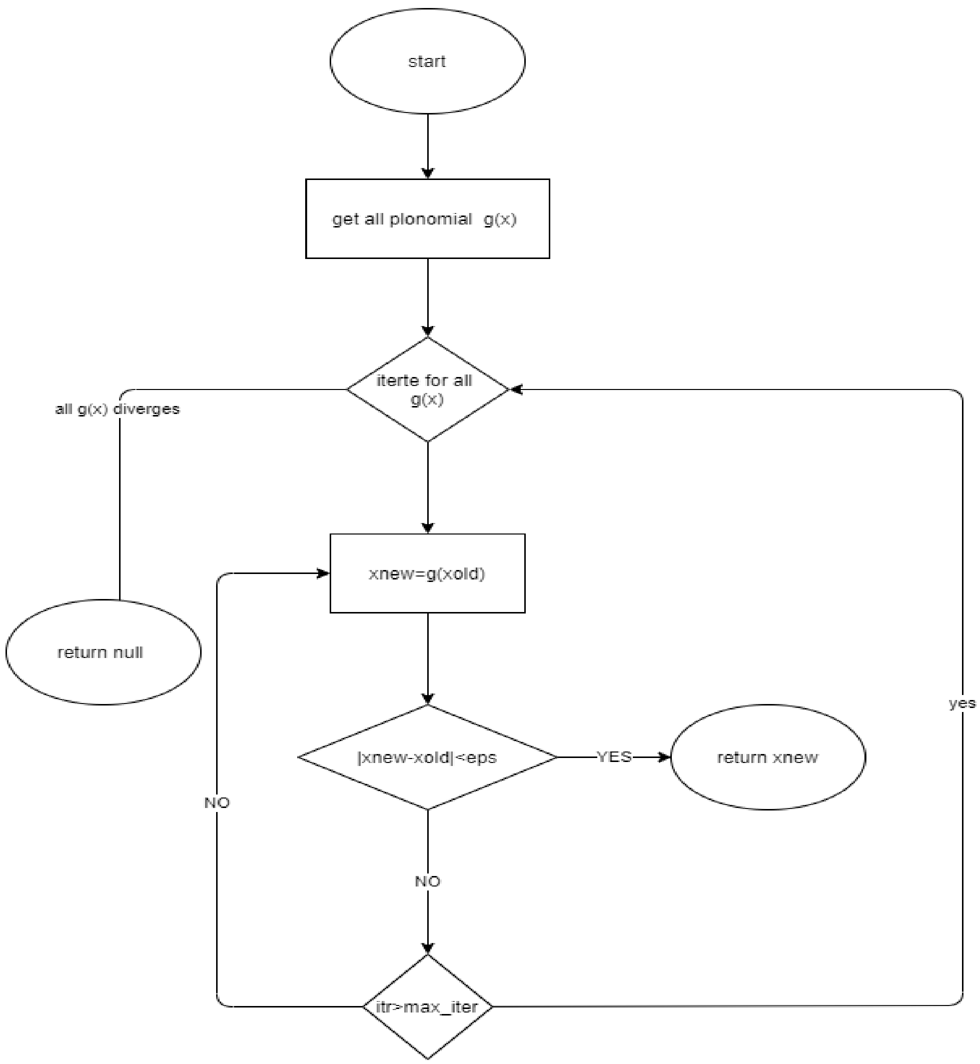
i. Bisection



ii. False-position



iii. Fixed point



g(x):

We get all polynomial terms and use it to make a corresponding g(x)

For example:

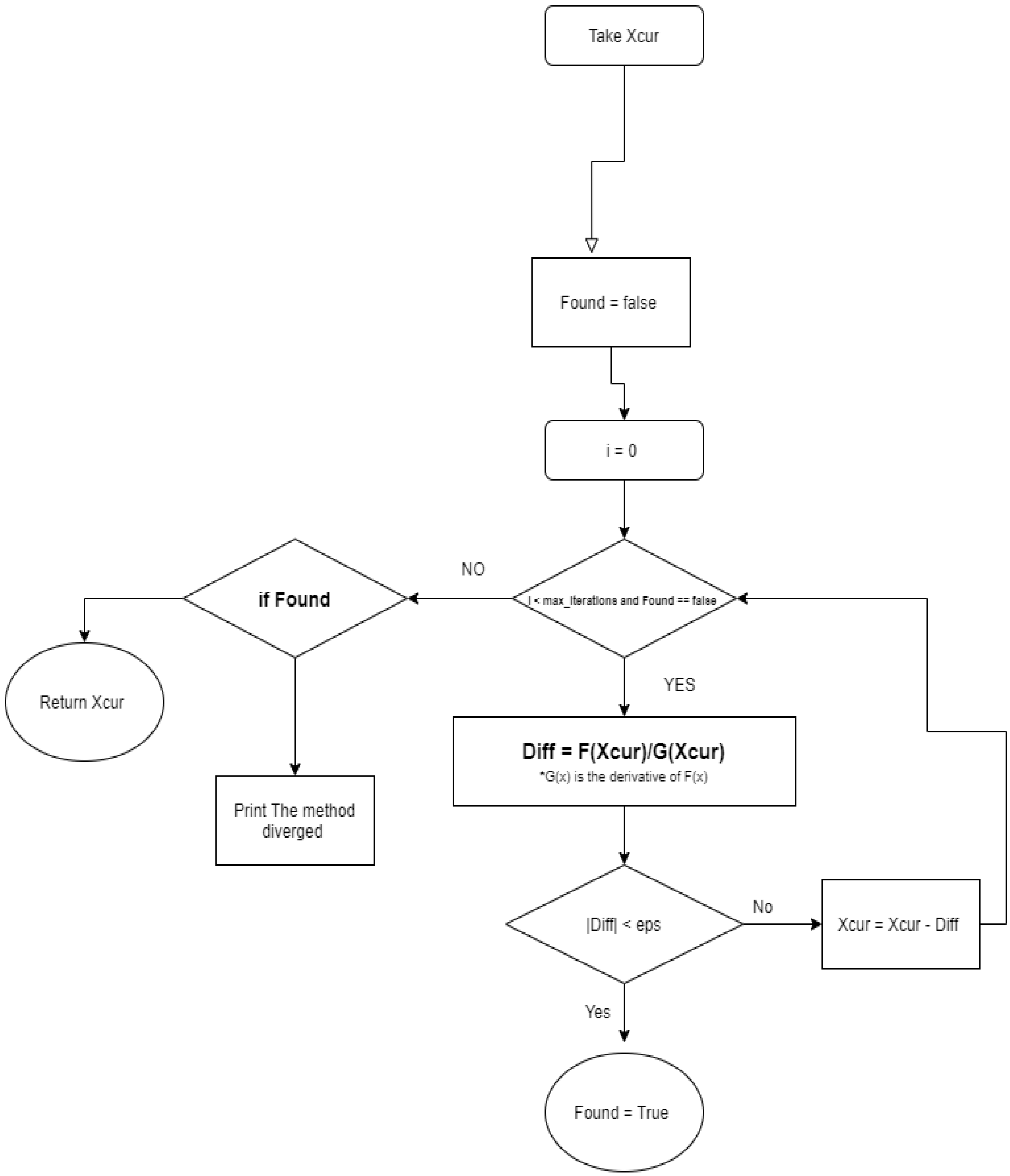
-7*x^3 +x^2+10x+13

g(x) are :

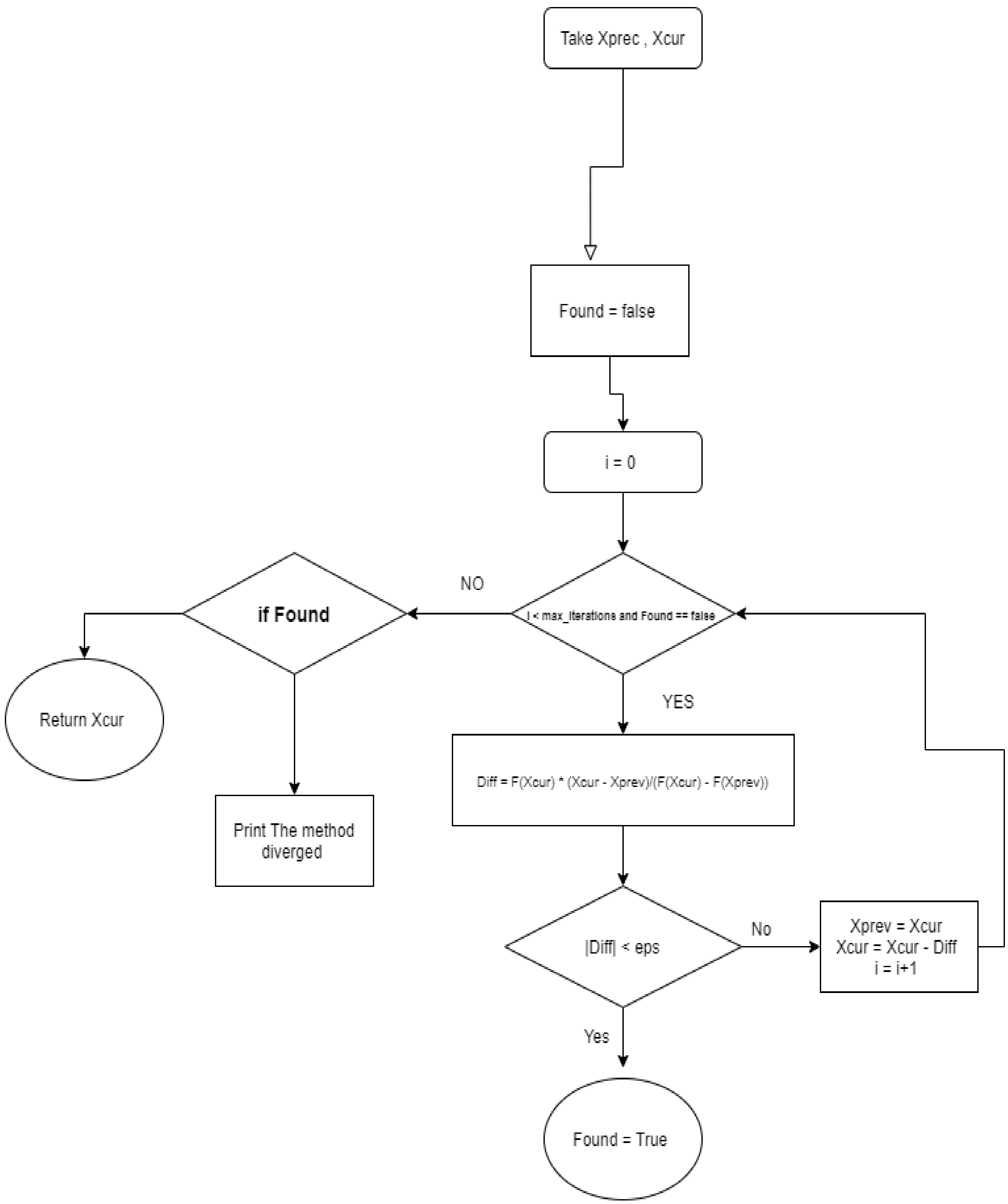
g(x)= ((x^2+10x+13)/-7)^1/3
g(x)= (-(-7*x^3+10x+13))^1/2
g(x)= -(-7*x^3 +x^2+13)/10
g(x)=x-(-7*x^3 +x^2+10x+13)

Then the answer converge from the first g(x)
To 1.667487796411278 (Correct)

iv. Newton-Raphson



v. Secant



Take Xprec , Xcur
Found = false
i = 0
i < max_iterations and Found == false
NO
if Found
Return Xcur
Print The method diverged
YES
Diff = F(Xcur) * (Xcur - Xprev)/(F(Xcur) - F(Xprev))
|Diff| < eps
No
Xprev = Xcur
Xcur = Xcur - Diff
i = i+1
Yes
Found = True

c. Analysis for the behavior

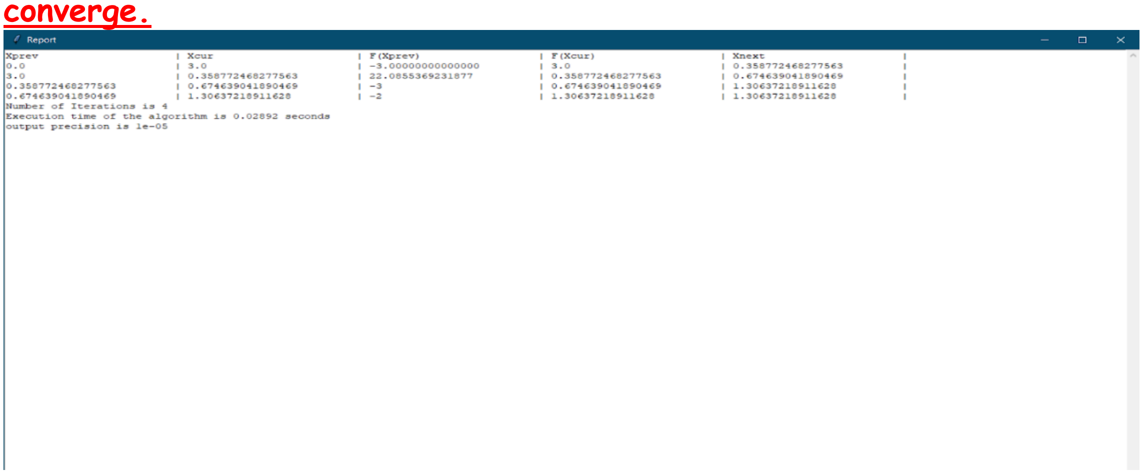
- **Example 1:**

Input : exp(x)+x**2-x-4.

xl = 0, xu = 5 (for bracketing methods and secant)
Max iteration : 100
Eps : .00001

method	Execution time (s)
Bisection	0.03169
False Position	0.07284
Fixed Point	diverge
Newton Raphson	0.02394
Secant	Division by zero

Note: if we changed the initial guesses at secant method it may converge.


Report

Xprev	Xcur	F(Xprev)	F(Xcur)	Xnext
0.0	3.0	-3.00000000000000	3.0	0.358772468277563
3.0	0.358772468277563	22.0855369231877	0.358772468277563	0.674639041890469
0.358772468277563	0.674639041890469	-3	0.674639041890469	1.30637218911628
0.674639041890469	1.30637218911628	-2	1.30637218911628	1.30637218911628

Number of Iterations is 4
Execution time of the algorithm is 0.02892 seconds
output precision is 1e-05

- **Example 2:**

Input :

-2*x**2+10.

Xl = 1 , xu = 10 (for bracketing methods)

Initial guess = 1

Max iteration = 50

Eps = .00001

method	Execution time (s)
Bisection	0.01374
False Position	0.01557
Fixed Point	0.00100
Newton Raphson	0.00694
Secant	0.00153

Example 3:

Input :

x2-4.**

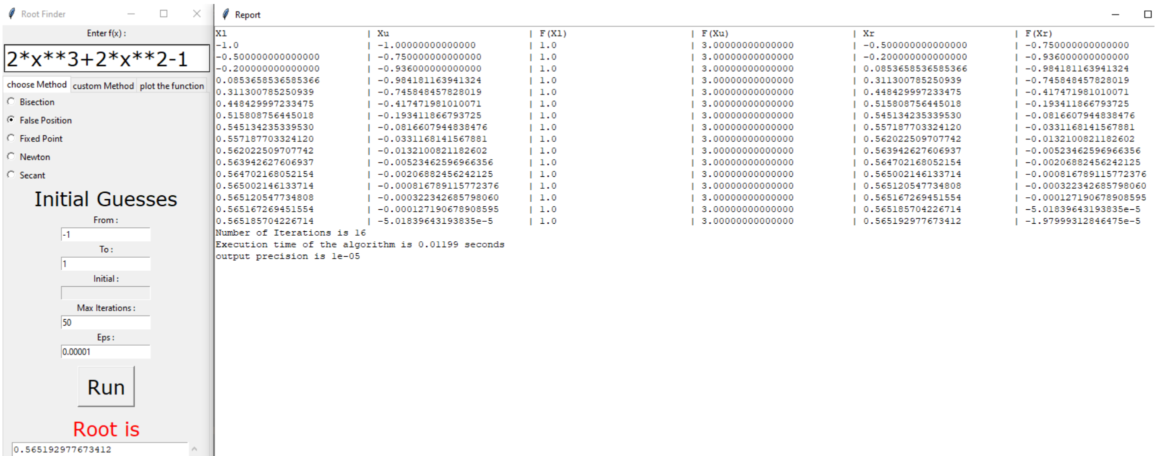
Xl = 0, xu = 4 (for bracketing methods)
Initial guess : 1.5
Max iteration : 50

Eps : 0.00005

method	Execution time (s)
Bisection	0.00994
False Position	0.01097
Fixed Point	0.00099
Newton Raphson	0.00997
Secant	0.00689

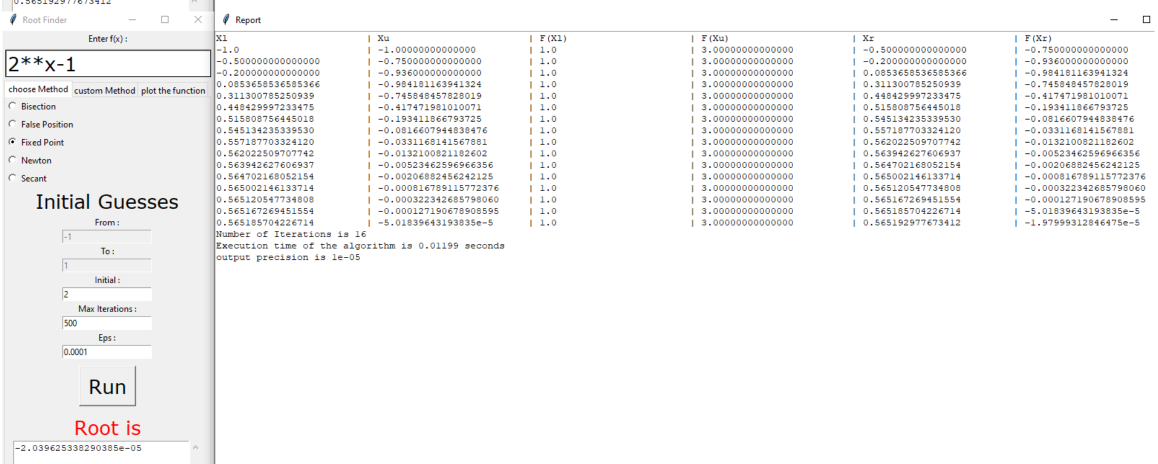
*the screens of this example is provided below

d. GUI and simple runs



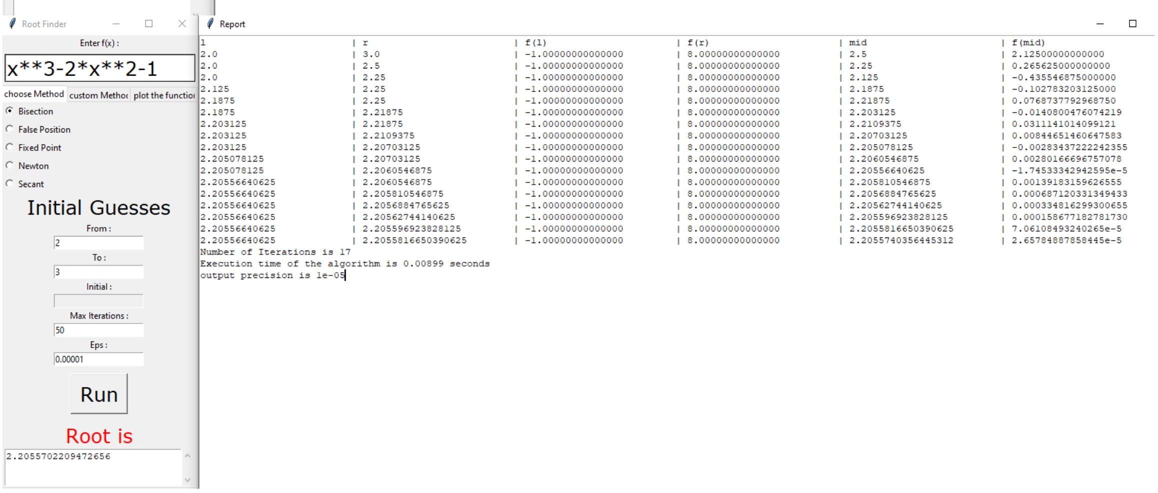
Xl	Xu	F(Xl)	F(Xu)	Xr	F(Xr)
-1.0	-1.00000000000000	1.0	3.00000000000000	-0.500000000000000	-0.750000000000000
-0.500000000000000	-0.750000000000000	1.0	3.00000000000000	-0.200000000000000	-0.936000000000000
-0.200000000000000	-0.936000000000000	1.0	3.00000000000000	0.0853658536585366	-0.984181163941324
0.0853658536585366	-0.984181163941324	1.0	3.00000000000000	0.311300785250939	-0.745848457828019
0.311300785250939	-0.745848457828019	1.0	3.00000000000000	0.448429997233475	-0.417471981010071
0.448429997233475	-0.417471981010071	1.0	3.00000000000000	0.515808756445018	-0.193411866793725
0.515808756445018	-0.193411866793725	1.0	3.00000000000000	0.545134235339530	-0.0816607944838476
0.545134235339530	-0.0816607944838476	1.0	3.00000000000000	0.557187703324120	-0.0331168141567881
0.557187703324120	-0.0331168141567881	1.0	3.00000000000000	0.562022509707742	-0.0132100821182602
0.562022509707742	-0.0132100821182602	1.0	3.00000000000000	0.563942627606937	-0.00523462596966356
0.563942627606937	-0.00523462596966356	1.0	3.00000000000000	0.564702168052154	-0.00206882456242125
0.564702168052154	-0.00206882456242125	1.0	3.00000000000000	0.565002146133714	-0.000816789115772376
0.565002146133714	-0.000816789115772376	1.0	3.00000000000000	0.565120547734808	-0.000322342685798060
0.565120547734808	-0.000322342685798060	1.0	3.00000000000000	0.565167269451554	-0.000127190678908595
0.565167269451554	-0.000127190678908595	1.0	3.00000000000000	0.565185704226714	-5.01839643193835e-5
0.565185704226714	-5.01839643193835e-5	1.0	3.00000000000000	0.565192977673412	-1.97999312846475e-5

Number of Iterations is 16
Execution time of the algorithm is 0.01199 seconds
output precision is 1e-05



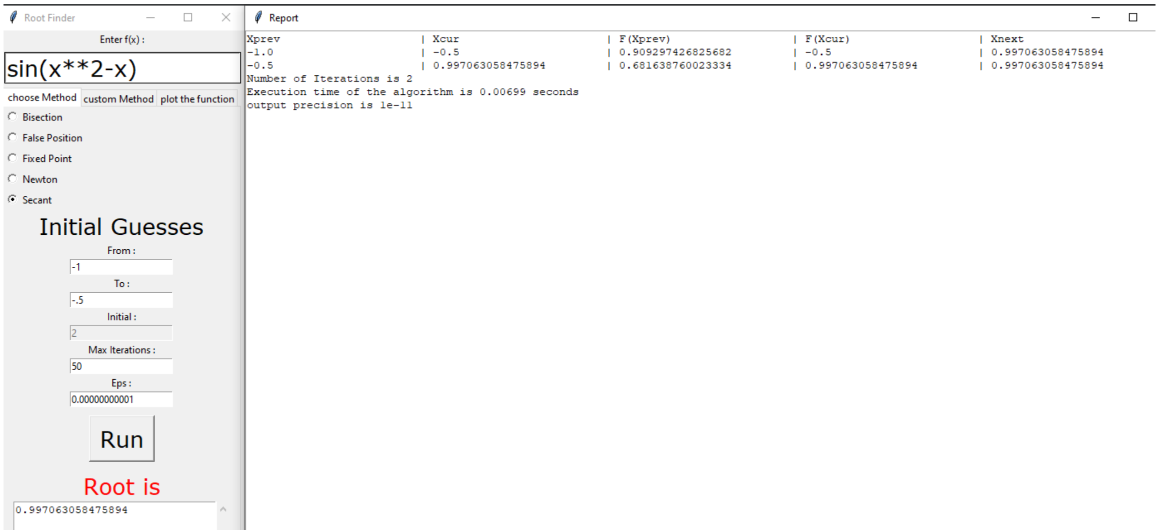
Xl	Xu	F(Xl)	F(Xu)	Xr	F(Xr)
-1.0	-1.00000000000000	1.0	3.00000000000000	-0.500000000000000	-0.750000000000000
-0.500000000000000	-0.750000000000000	1.0	3.00000000000000	-0.200000000000000	-0.936000000000000
-0.200000000000000	-0.936000000000000	1.0	3.00000000000000	0.0853658536585366	-0.984181163941324
0.0853658536585366	-0.984181163941324	1.0	3.00000000000000	0.311300785250939	-0.745848457828019
0.311300785250939	-0.745848457828019	1.0	3.00000000000000	0.448429997233475	-0.417471981010071
0.448429997233475	-0.417471981010071	1.0	3.00000000000000	0.515808756445018	-0.193411866793725
0.515808756445018	-0.193411866793725	1.0	3.00000000000000	0.545134235339530	-0.0816607944838476
0.545134235339530	-0.0816607944838476	1.0	3.00000000000000	0.557187703324120	-0.0331168141567881
0.557187703324120	-0.0331168141567881	1.0	3.00000000000000	0.562022509707742	-0.0132100821182602
0.562022509707742	-0.0132100821182602	1.0	3.00000000000000	0.563942627606937	-0.00523462596966356
0.563942627606937	-0.00523462596966356	1.0	3.00000000000000	0.564702168052154	-0.00206882456242125
0.564702168052154	-0.00206882456242125	1.0	3.00000000000000	0.565002146133714	-0.000816789115772376
0.565002146133714	-0.000816789115772376	1.0	3.00000000000000	0.565120547734808	-0.000322342685798060
0.565120547734808	-0.000322342685798060	1.0	3.00000000000000	0.565167269451554	-0.000127190678908595
0.565167269451554	-0.000127190678908595	1.0	3.00000000000000	0.565185704226714	-5.01839643193835e-5
0.565185704226714	-5.01839643193835e-5	1.0	3.00000000000000	0.565192977673412	-1.97999312846475e-5

Number of Iterations is 16
Execution time of the algorithm is 0.01199 seconds
output precision is 1e-05

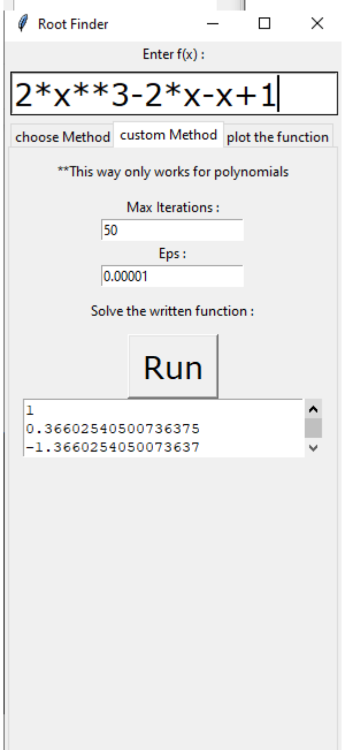


l	r	f(l)	f(r)	mid	f(mid)
2.0	3.0	-1.00000000000000	8.00000000000000	2.5	2.12500000000000
2.0	2.5	-1.00000000000000	8.00000000000000	2.25	0.265625000000000
2.0	2.25	-1.00000000000000	8.00000000000000	2.125	-0.435546875000000
2.125	2.25	-1.00000000000000	8.00000000000000	2.1875	-0.102783203125000
2.1875	2.25	-1.00000000000000	8.00000000000000	2.21875	0.0768737792968750
2.1875	2.21875	-1.00000000000000	8.00000000000000	2.203125	-0.0140800476074219
2.203125	2.21875	-1.00000000000000	8.00000000000000	2.2109375	0.0311141014099121
2.203125	2.2109375	-1.00000000000000	8.00000000000000	2.20703125	0.00844651460647583
2.203125	2.20703125	-1.00000000000000	8.00000000000000	2.205078125	-0.00283437222242355
2.205078125	2.20703125	-1.00000000000000	8.00000000000000	2.2060546875	0.00280166696757078
2.205078125	2.2060546875	-1.00000000000000	8.00000000000000	2.20556640625	-1.74533342942595e-5
2.20556640625	2.2060546875	-1.00000000000000	8.00000000000000	2.205810546875	0.00139183159626555
2.20556640625	2.205810546875	-1.00000000000000	8.00000000000000	2.2056884765625	0.000687120331349433
2.20556640625	2.2056884765625	-1.00000000000000	8.00000000000000	2.20562744140625	0.000334816299300655
2.20556640625	2.20562744140625	-1.00000000000000	8.00000000000000	2.205596923828125	0.000158677182781730
2.20556640625	2.205596923828125	-1.00000000000000	8.00000000000000	2.2055816650390625	7.06108493240265e-5
2.20556640625	2.2055816650390625	-1.00000000000000	8.00000000000000	2.2055740356445312	2.65784887858445e-5

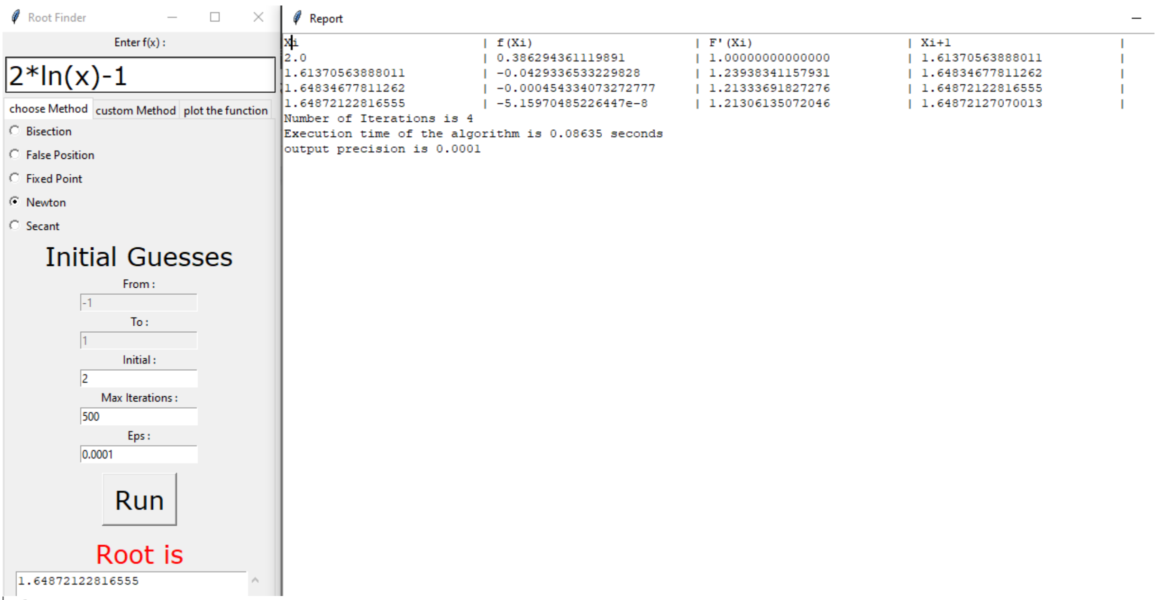
Number of Iterations is 17
Execution time of the algorithm is 0.00899 seconds
output precision is 1e-05



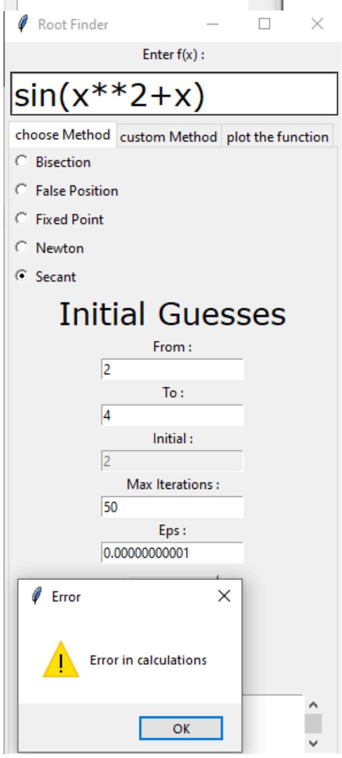
Root Finder
Enter f(x) :
sin(x**2-x)
choose Method
custom Method
plot the function
Bisection
False Position
Fixed Point
Newton
Secant
Initial Guesses
From :
-1
To :
-.5
Initial :
2
Max Iterations :
50
Eps :
0.00000000001
Run
Root is
0.997063058475894
Report
Xprev | Xcur | F(Xprev) | F(Xcur) | Xnext
-1.0 | -0.5 | 0.909297426825682 | -0.5 | 0.997063058475894
-0.5 | 0.997063058475894 | 0.681638760023334 | 0.997063058475894 | 0.997063058475894
Number of Iterations is 2
Execution time of the algorithm is 0.00699 seconds
output precision is 1e-11



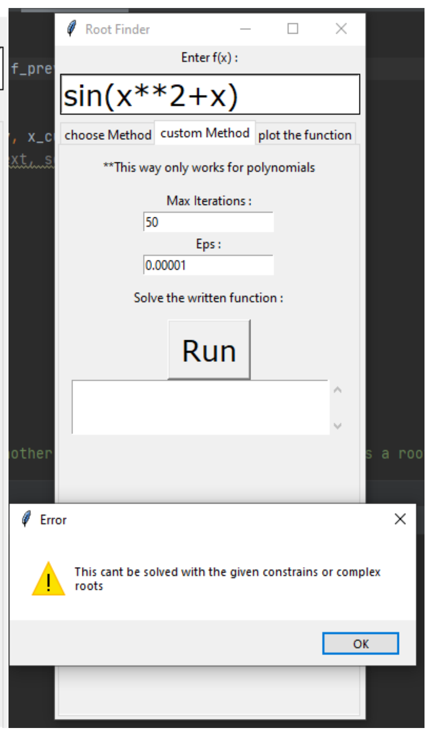
Root Finder
Enter f(x) :
2*x**3-2*x-x+1
choose Method
custom Method
plot the function
**This way only works for polynomials
Max Iterations :
50
Eps :
0.00001
Solve the written function :
Run
1
0.36602540500736375
-1.3660254050073637



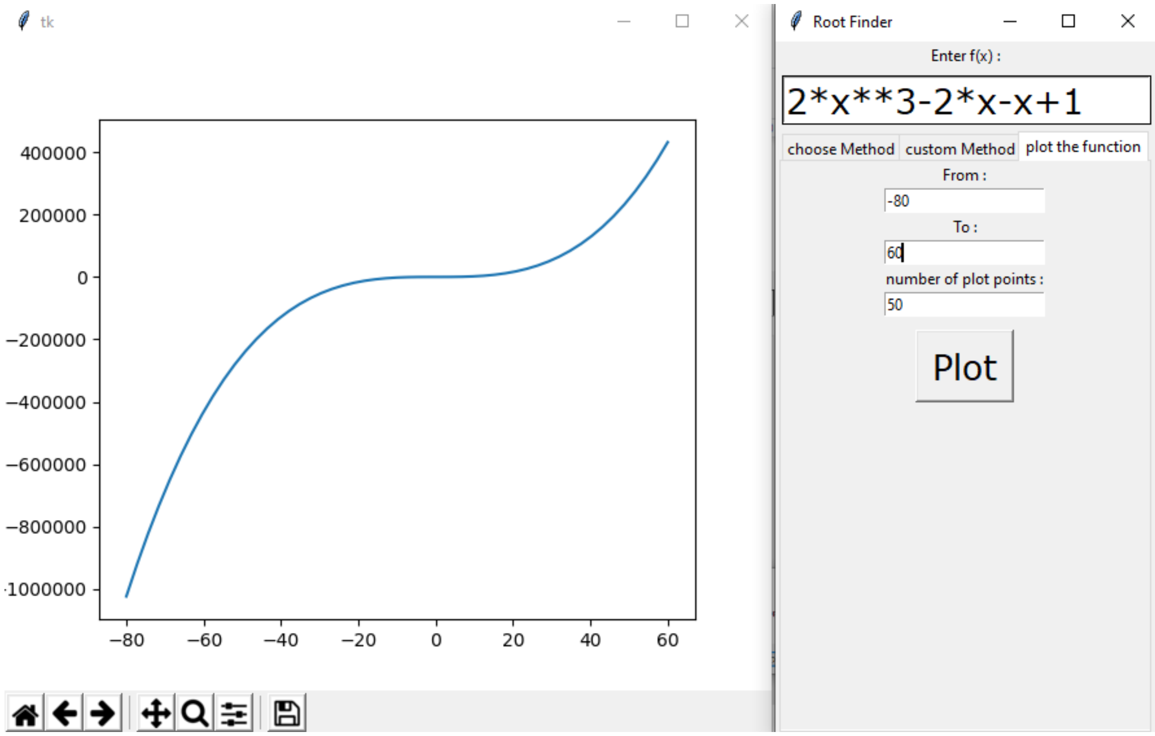
Root Finder
Enter f(x) :
2*ln(x)-1
choose Method
custom Method
plot the function
Bisection
False Position
Fixed Point
Newton
Secant
Initial Guesses
From :
-1
To :
1
Initial :
2
Max Iterations :
500
Eps :
0.0001
Run
Root is
1.64872122816555
Report
Xi | f(Xi) | F'(Xi) | Xi+1 |
2.0 | 0.386294361119891 | 1.00000000000000 | 1.61370563888011 |
1.61370563888011 | -0.0429336533229828 | 1.23938341157931 | 1.64834677811262 |
1.64834677811262 | -0.000454334073272777 | 1.21333691827276 | 1.64872122816555 |
1.64872122816555 | -5.15970485226447e-8 | 1.21306135072046 | 1.64872127070013 |
Number of Iterations is 4
Execution time of the algorithm is 0.08635 seconds
output precision is 0.0001



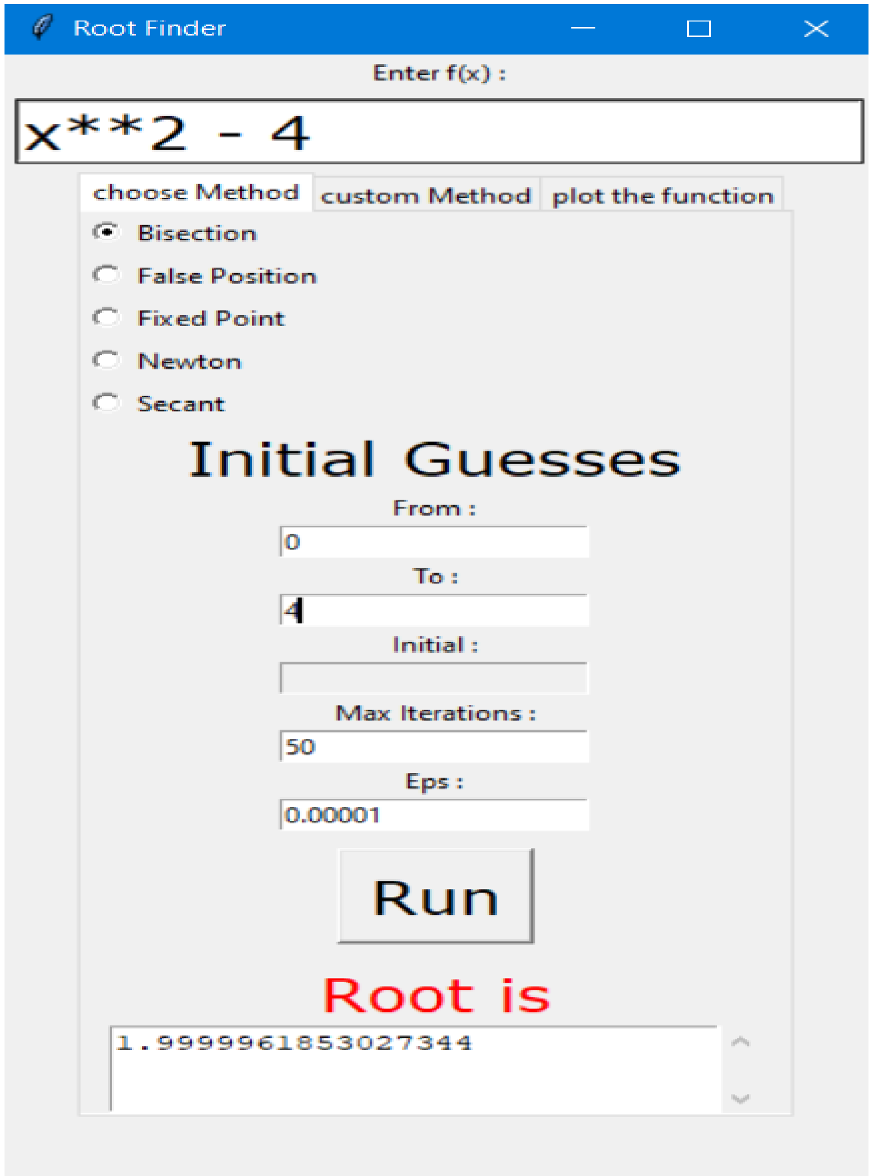
Root Finder
Enter f(x) :
sin(x**2+x)
choose Method
custom Method
plot the function
Bisection
False Position
Fixed Point
Newton
Secant
Initial Guesses
From :
2
To :
4
Initial :
2
Max Iterations :
50
Eps :
0.00000000001
Error
Error in calculations
OK



Root Finder
Enter f(x) :
sin(x**2+x)
choose Method
custom Method
plot the function
**This way only works for polynomials
Max Iterations :
50
Eps :
0.00001
Solve the written function :
Run
Error
This cant be solved with the given constrains or complex roots
OK


tk
400000
200000
0
−200000
−400000
−600000
−800000
-1000000
−80
−60
−40
−20
0
20
40
60
Root Finder
Enter f(x) :
2*x**3-2*x-x+1
choose Method
custom Method
plot the function
From :
-80
To :
60
number of plot points :
50
Plot

1-Comprehensive



Example:

Report

l	r	f(l)	f(r)	mid	f(mid)
0.0	4.0	-4.00000000000000	12.0000000000000	2.0	0
0.0	2.0	-4.00000000000000	12.0000000000000	1.0	-3.00000000000000
1.0	2.0	-4.00000000000000	12.0000000000000	1.5	-1.75000000000000
1.5	2.0	-4.00000000000000	12.0000000000000	1.75	-0.937500000000000
1.75	2.0	-4.00000000000000	12.0000000000000	1.875	-0.484375000000000
1.875	2.0	-4.00000000000000	12.0000000000000	1.9375	-0.246093750000000
1.9375	2.0	-4.00000000000000	12.0000000000000	1.96875	-0.124023437500000
1.96875	2.0	-4.00000000000000	12.0000000000000	1.984375	-0.0622558593750000
1.984375	2.0	-4.00000000000000	12.0000000000000	1.9921875	-0.0311889648437500
1.9921875	2.0	-4.00000000000000	12.0000000000000	1.99609375	-0.0156097412109375
1.99609375	2.0	-4.00000000000000	12.0000000000000	1.998046875	-0.00780868530273438
1.998046875	2.0	-4.00000000000000	12.0000000000000	1.9990234375	-0.00390529632568359
1.9990234375	2.0	-4.00000000000000	12.0000000000000	1.99951171875	-0.00195288658142090
1.99951171875	2.0	-4.00000000000000	12.0000000000000	1.999755859375	-0.000976502895355225
1.999755859375	2.0	-4.00000000000000	12.0000000000000	1.9998779296875	-0.000488266348838806
1.9998779296875	2.0	-4.00000000000000	12.0000000000000	1.99993896484375	-0.000244136899709702
1.99993896484375	2.0	-4.00000000000000	12.0000000000000	1.999969482421875	-0.000122069381177425
1.999969482421875	2.0	-4.00000000000000	12.0000000000000	1.9999847412109375	-6.10349234193563e-5
1.9999847412109375	2.0	-4.00000000000000	12.0000000000000	1.9999923706054688	-3.05175199173391e-5

Number of Iterations is 19
Execution time of the algorithm is 0.00994 seconds
output precision is 1e-05

Xl	Xu	F(Xl)	F(Xu)	Xr	F(Xr)
0.0	-4.00000000000000	4.0	12.0000000000000	1.00000000000000	-3.00000000000000
1.00000000000000	-3.00000000000000	4.0	12.0000000000000	1.60000000000000	-1.44000000000000
1.60000000000000	-1.44000000000000	4.0	12.0000000000000	1.85714285714286	-0.551020408163265
1.85714285714286	-0.551020408163265	4.0	12.0000000000000	1.95121951219512	-0.192742415229031
1.95121951219512	-0.192742415229031	4.0	12.0000000000000	1.98360655737705	-0.0653050255307712
1.98360655737705	-0.0653050255307712	4.0	12.0000000000000	1.99452054794521	-0.0218877838243574
1.99452054794521	-0.0218877838243574	4.0	12.0000000000000	1.99817184643510	-0.00730927211414167
1.99817184643510	-0.00730927211414167	4.0	12.0000000000000	1.99939042974703	-0.00243790943599409
1.99939042974703	-0.00243790943599409	4.0	12.0000000000000	1.99979678927047	-0.000812801623504278
1.99979678927047	-0.000812801623504278	4.0	12.0000000000000	1.99993226079594	-0.000270952227657162
1.99993226079594	-0.000270952227657162	4.0	12.0000000000000	1.99997742001039	-9.03194485961656e-5
1.99997742001039	-9.03194485961656e-5	4.0	12.0000000000000	1.99999247330847	-3.01067094663487e-5
1.99999247330847	-3.01067094663487e-5	4.0	12.0000000000000	1.99999749109968	-1.00355950012343e-5

Number of Iterations is 13
Execution time of the algorithm is 0.01097 seconds
output precision is 1e-05

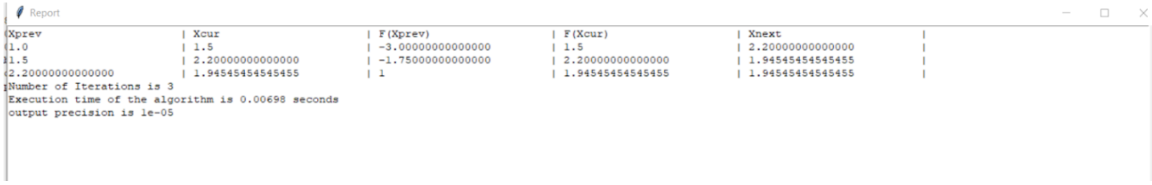
i	X
1	2.0
2	2.0

Number of Iterations is 2
Execution time of the algorithm is 0.00099 seconds
output precision is 1e-05

Report

Xi	f(Xi)	F'(Xi)	Xi+1
1.5	-1.75000000000000	3.00000000000000	2.08333333333333
2.08333333333333	0.340277777777779	4.16666666666667	2.00166666666667
2.00166666666667	0.00666944444444351	4.00333333333333	2.00000069386622
2.00000069386622	2.77546537219564e-6	4.00000138773245	2.00000000000012

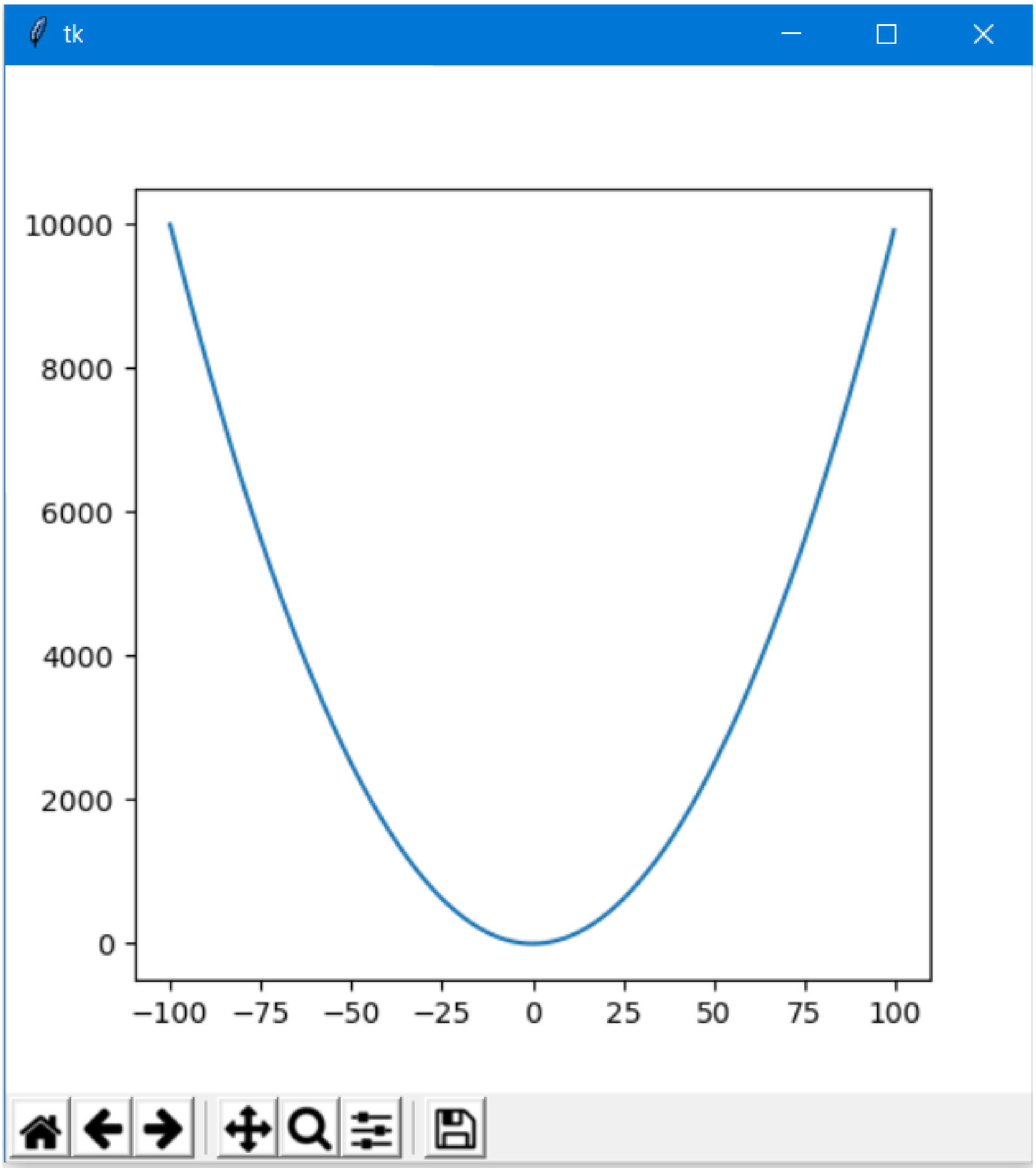
Number of Iterations is 4
Execution time of the algorithm is 0.00997 seconds
output precision is 1e-05



Report

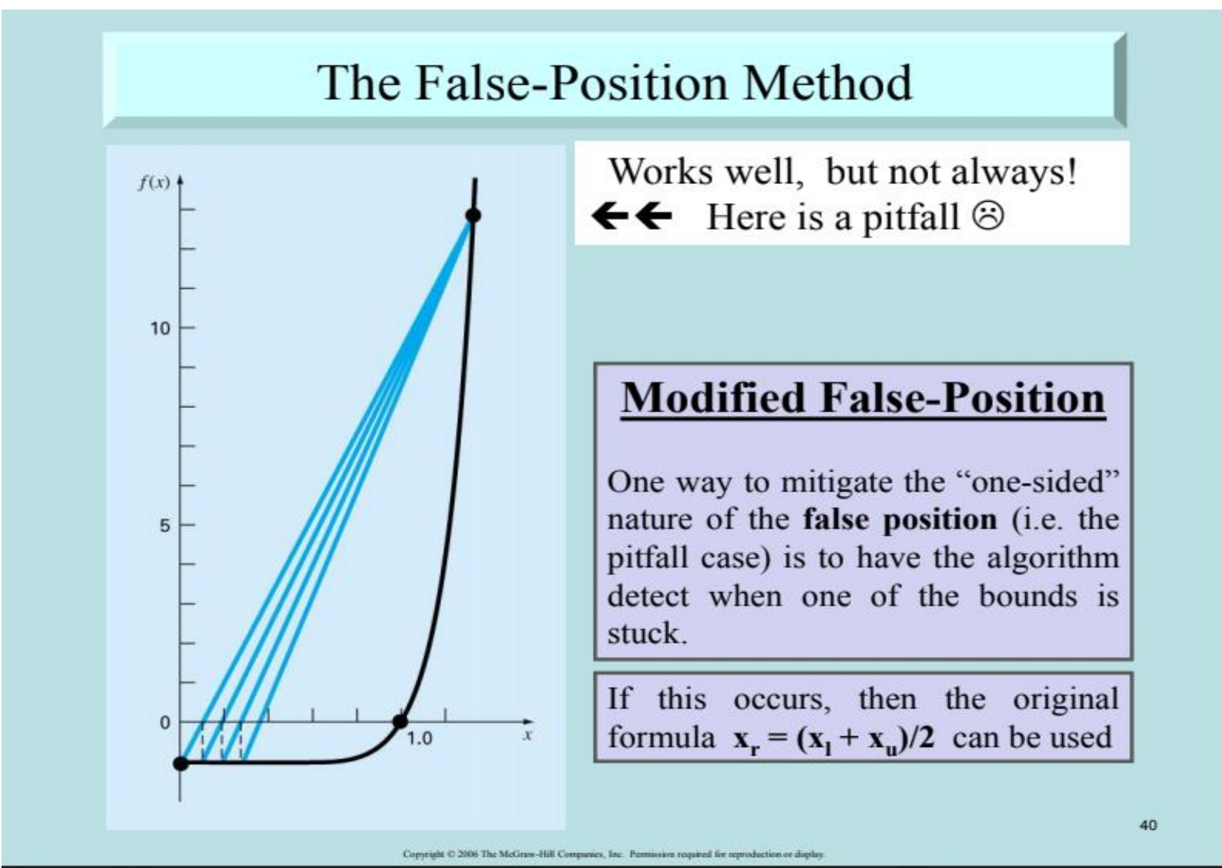
Xprev	Xcur	F(Xprev)	F(Xcur)	Xnext
1.0	1.5	-3.00000000000000	1.5	2.20000000000000
1.5	2.20000000000000	-1.75000000000000	2.20000000000000	1.94545454545455
2.20000000000000	1.94545454545455	1	1.94545454545455	1.94545454545455

Number of Iterations is 3
Execution time of the algorithm is 0.00698 seconds
output precision is 1e-05



Problematic functions

1- In the equation "Y = ex + x**2 - x - 4" , we noted that the number of iterations in false position Is almost twice the times of the number of iterations the bisection method took to get the find the root .**
A suggested solution to this problem, as stated in the reference, is to get the middle of the left and the right of the interval and make this midpoint the left or the right according to the equation and then we can implement the false position procedure.



2- Quadratic Convergence

In case of the equations that have multiplicity, the Newton method converges linearly.

-One possible solution to this problem is to modify the way we get the next point

$Y(x) = X_i - m * (f(x)/f'(x))$ where m is the multiplicity of the equation.