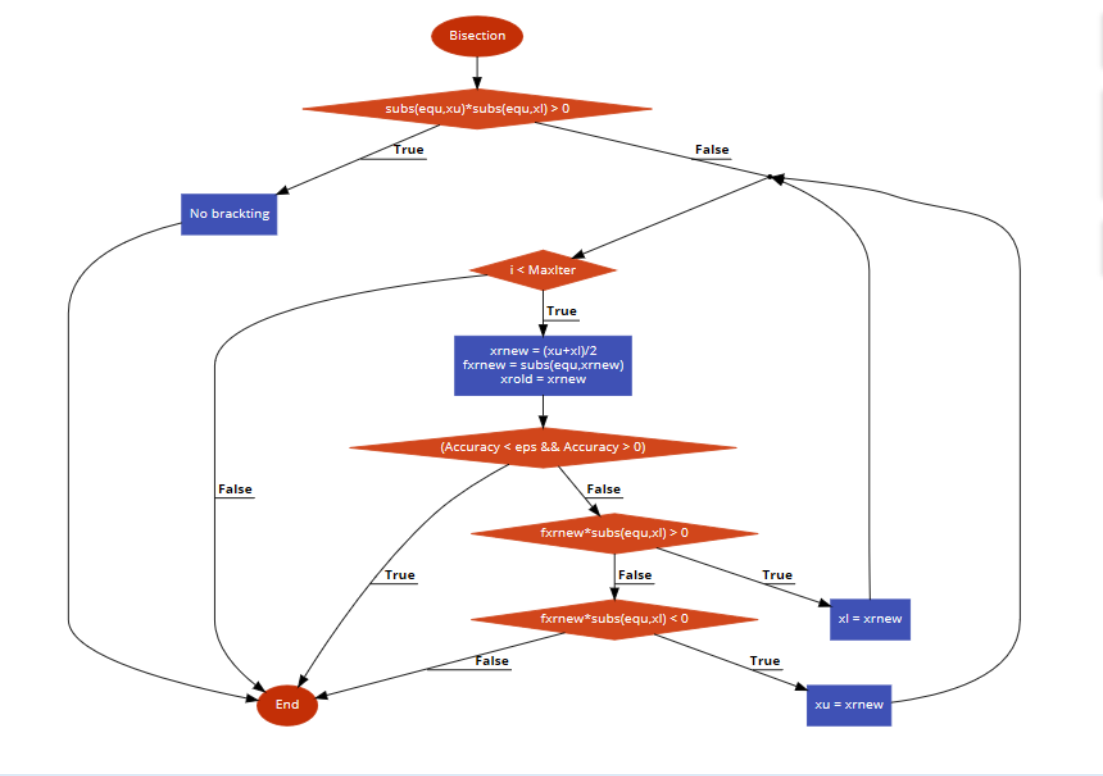
NUMERICAL ANALYSIS ASSIGNMENT 1 - PART 1

14 May 2019

FLOWCHART

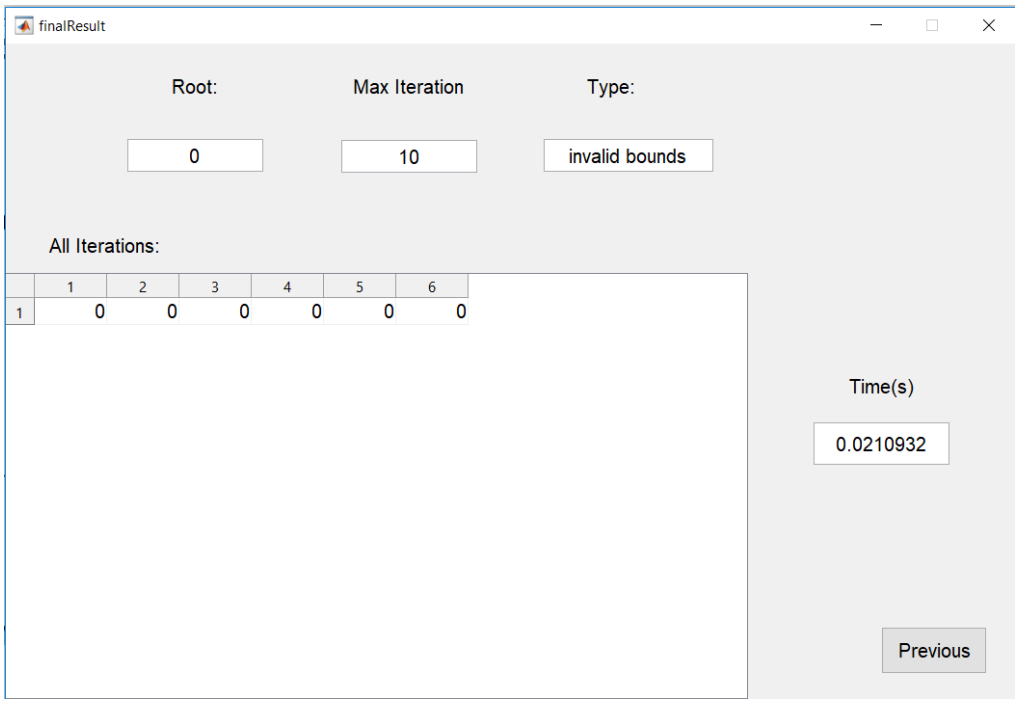
1. Bisection

Describe how bisection method works:

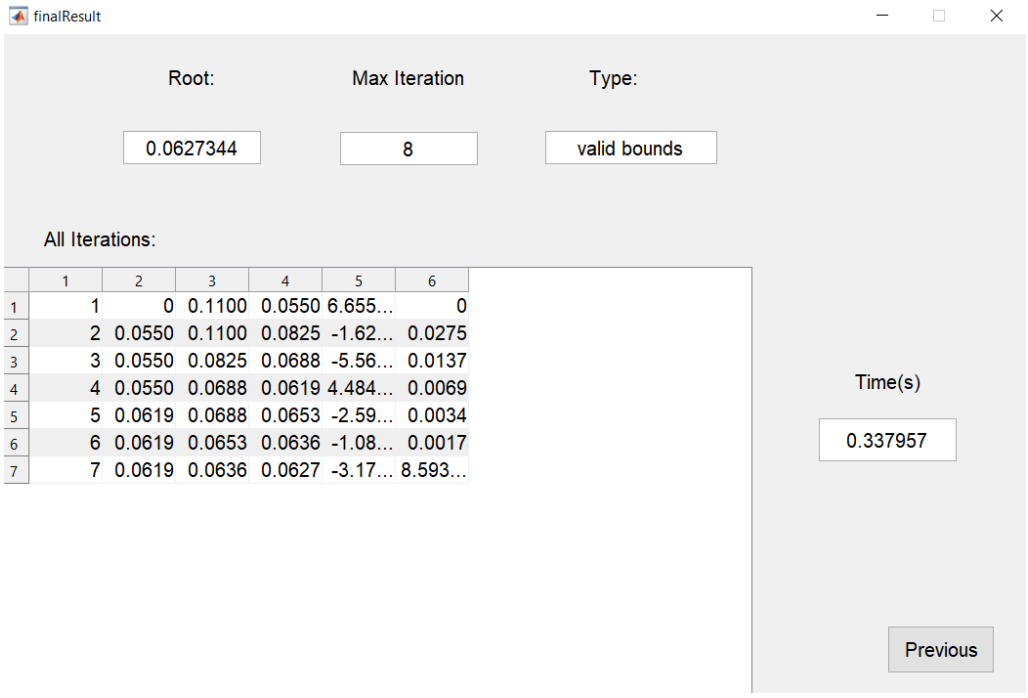


Analysis and conclusion for the behavior of Bisection Method:

- If xl and xu have an even numbers of roots in between then there is no bracketing.

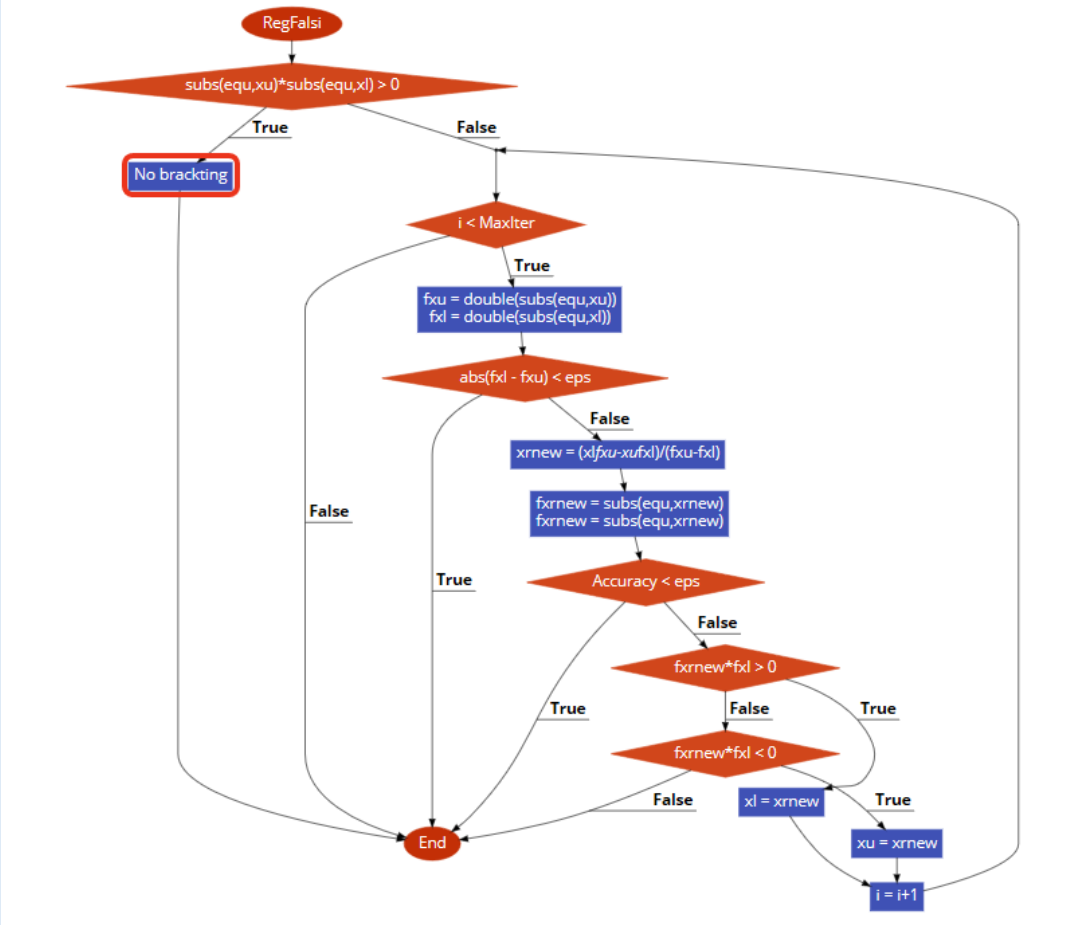


- If they have an odd number of roots in between then there is bracketing.



2. False-position

Describe how False-position method works:

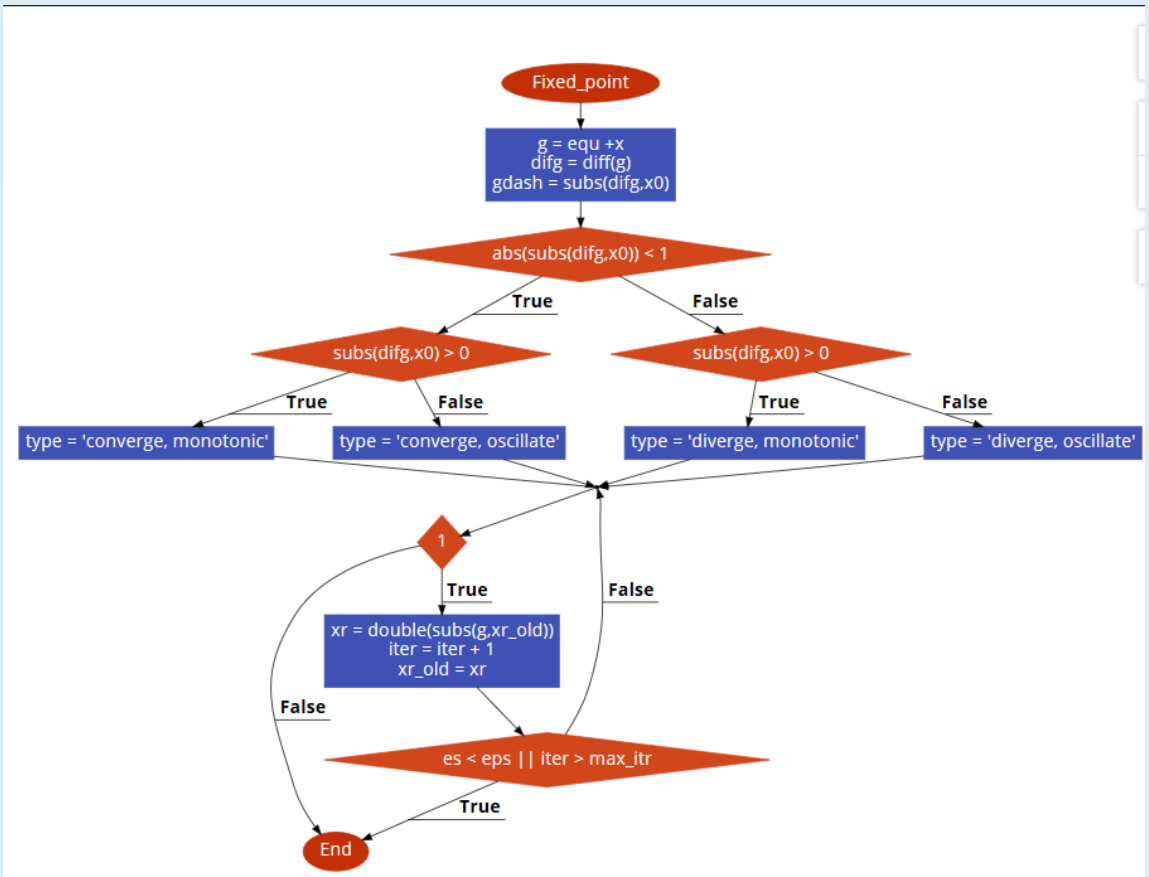


Analysis and conclusion for the behavior of False Position:

- If xl and xu have an even numbers of roots in between then there is no bracketing.
- If they have an odd number of roots in between then there is bracketing.

3. Fixed point

Describe how Fixed point method works:

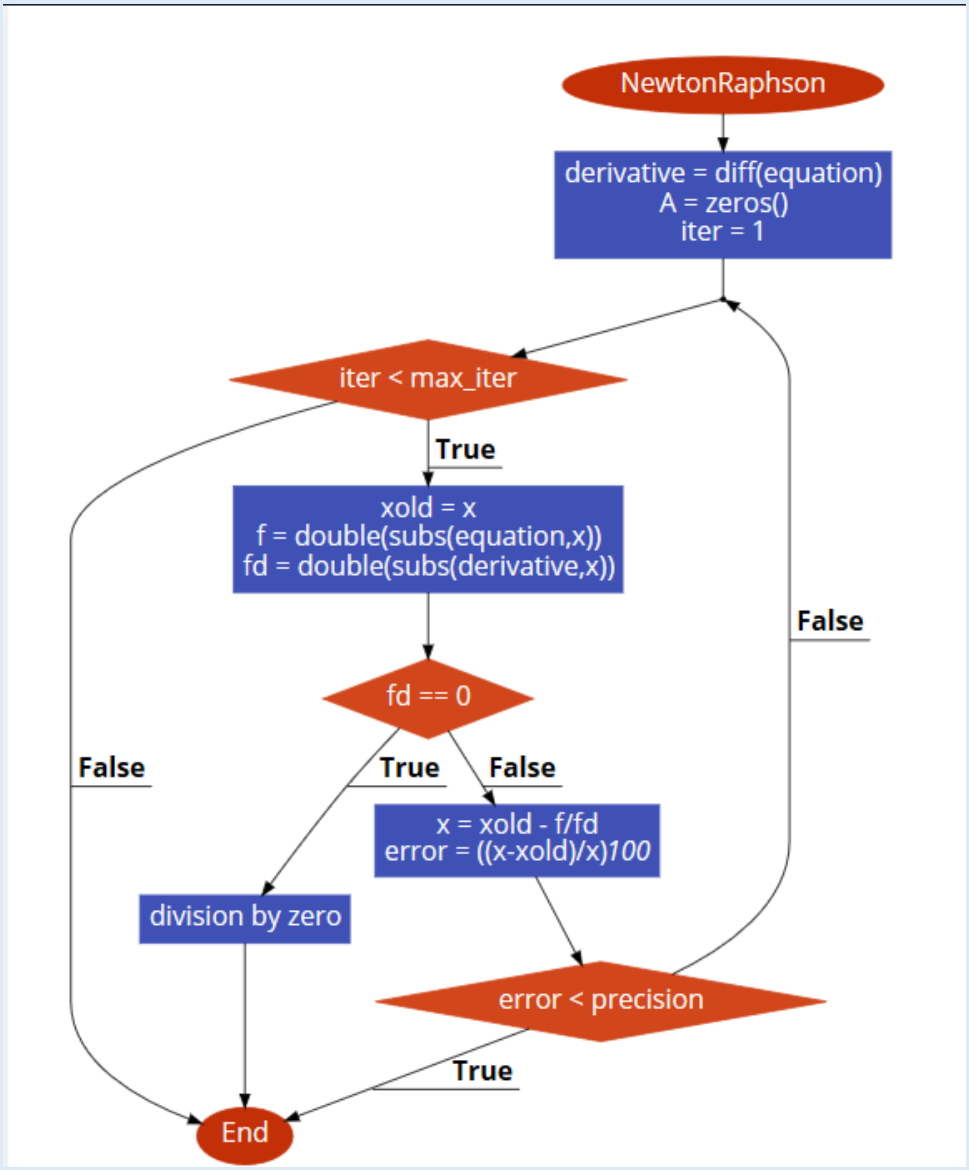


Analysis and conclusion for the behavior of Fixed_Point:

- If the absolute of differentiation of function g at x = x0 is smaller than 1 then function converge.
- If the absolute of differentiation of function g at x = x0 is bigger than 1 then function diverge.

4. Newton-Raphson

Describe how Newton Raphson method works:

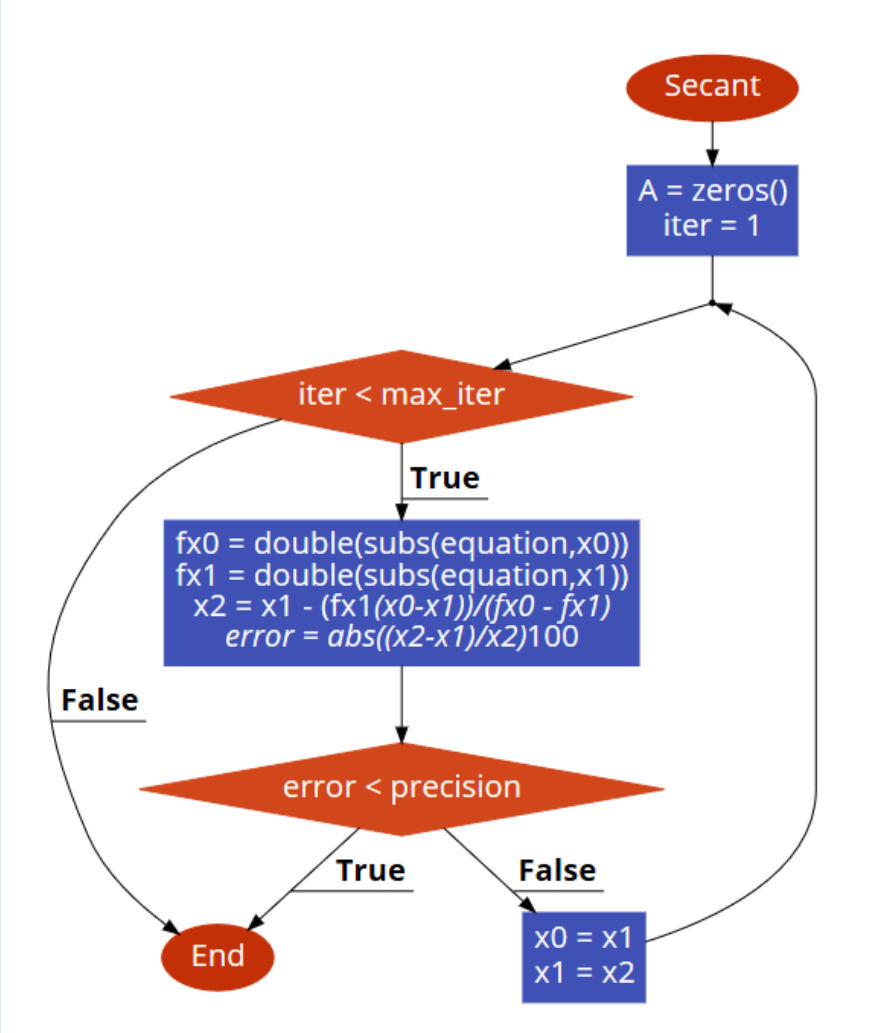


Analysis and conclusion for the behavior of Newton Raphson:

- If the differentiation of function at x = x0 is equal to 0 then it can't get the root.
- If the differentiation of function at x = x0 is not equal to 0 then it can get the root.

5. Secant

Describe how Secant method works:

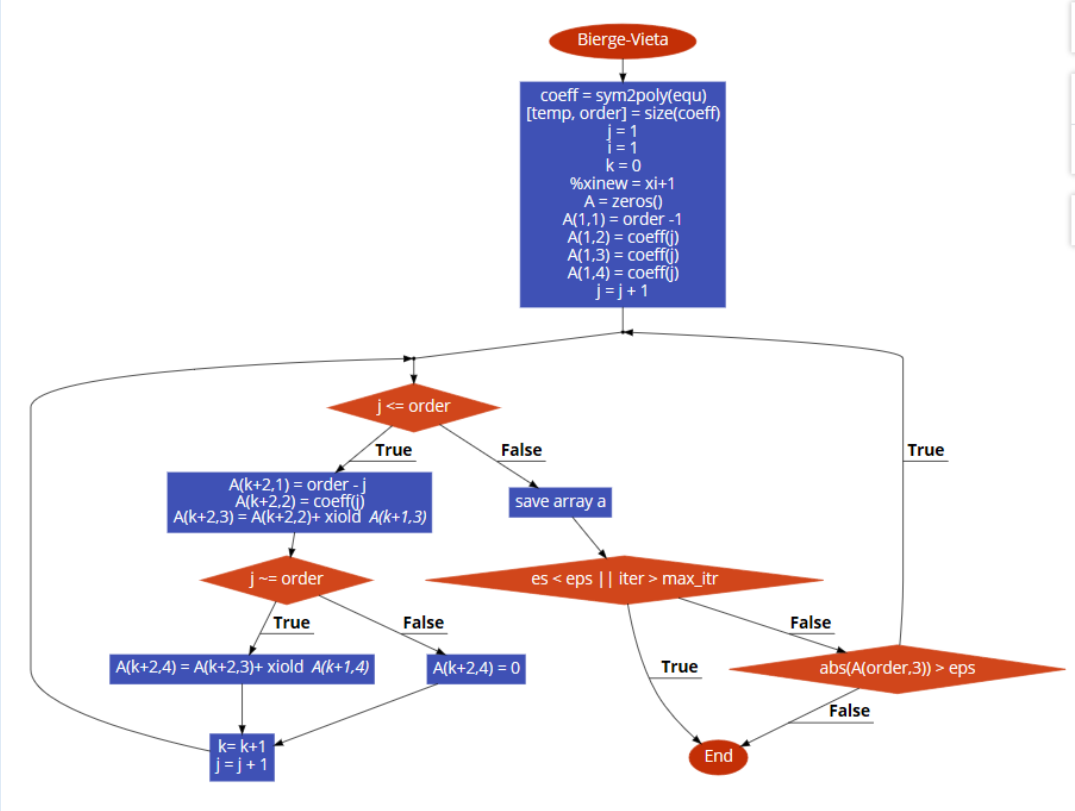


Analysis and conclusion for the behavior of Secant:

- We can't know at the beginning if it converges or diverge.
- It may converge like this example.
- It may diverge like this example.

6. Bierge-Vieta

Describe how Bierge-Vieta method works:



Analysis and conclusion for the behavior of Birge-Vieta:

- We can't know at the beginning if it converges or diverge.
- It may converge like this example.
- It may diverge like this example.

GENERAL ALGORITHM

1. Explain of general algorithm

The general algorithm first check if the equation is polynomial or not,

Case 1 (Polynomial of order n):

If the constant term of the equation is zero, then zero is a root of the equation. Otherwise, the initial guess is determined by the nth root of the constant term. Birge-Vieta is used to find the root by the calculated initial guess or its negative value, if diverged the interval between the initial guess, or its negative value as well, and zero is checked to be bracketing one of the

roots, if so Bisection method is used.

Case 2 (Transcendental):

A random constant is generated, this guess and four other guesses on some distance from its right and left are checked to converge on using one of these values as initial guess. If failed to converge, an interval from this initial guess is increased by some delta on both directions to bracket one of the roots, then False-Position method is used on this interval.

On both cases if no root is bracketed then Secant method is used instead.

2. Reason behind our decisions

Case 1:

Obviously, zero is a root for a zero-constant equation. If the constant is non-zero then its nth root is near one of the equation roots, so iterative methods as Birge-Vieta is supposed to converge. The value of the nth root lies at some point between all the roots of the equation, so the interval between zero and this point is expected to bracket odd numbers of root with high probability, otherwise these initial guesses are near some roots se Secant method is supposed to converge.

Case 2:

We try to get the random initial guess to some value that is supposed to converge on using Fixed_Point metod, otherwise we try to bracket one of the roots by increasing the bounds of the interval on both directions to ensure finding a root.

3. Data structures used

2-D arrays and 3-D arrays to save the steps of each method.

Vectors.

PROBLAMATIC FUNCTIONS

- **Newton Raphson:**

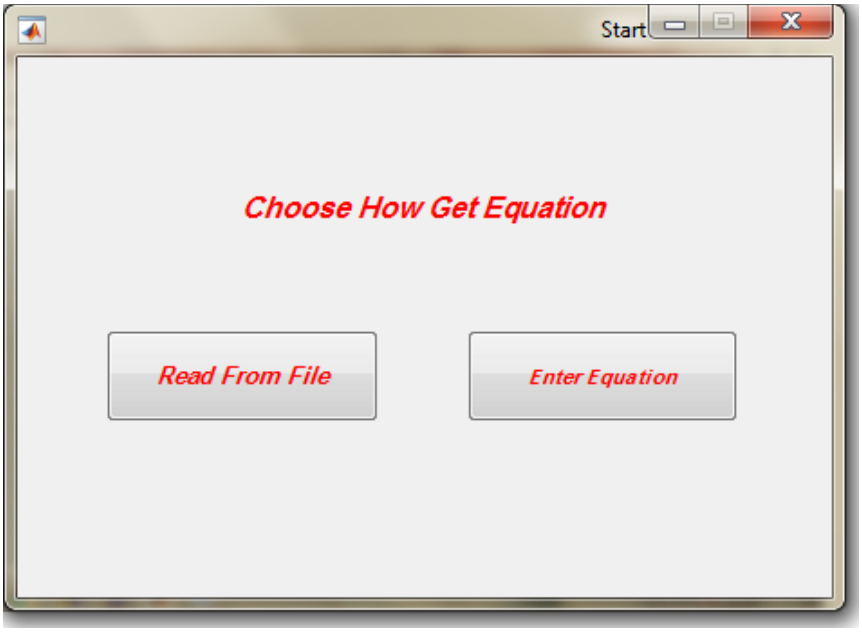
If the derivative of the function is zero or undefined at some point the algorithm will stop. This misbehavior is because of the division by the derivative on calculating the new root estimate.

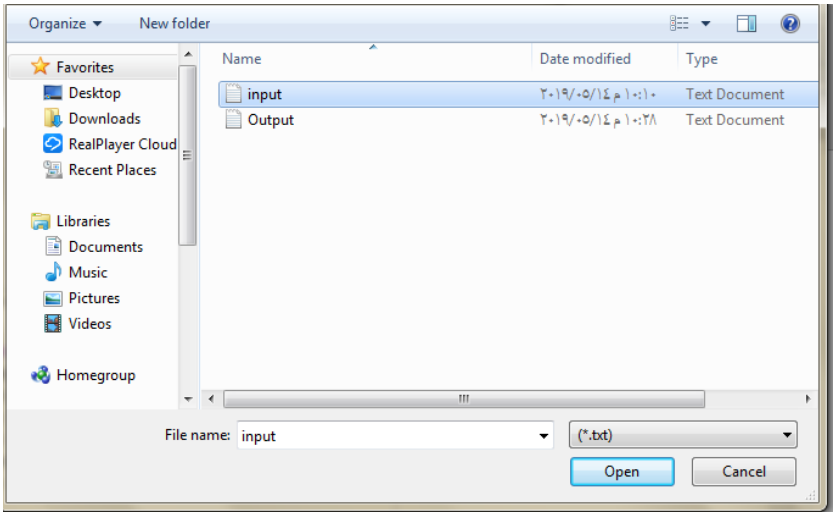
Trying to reformulate the formula to avoid division by zero may decrease the probability of this misbehavior.

Inflection points causes some iterative methods to diverge, also local maximums and minimums causes these methods to oscillate.

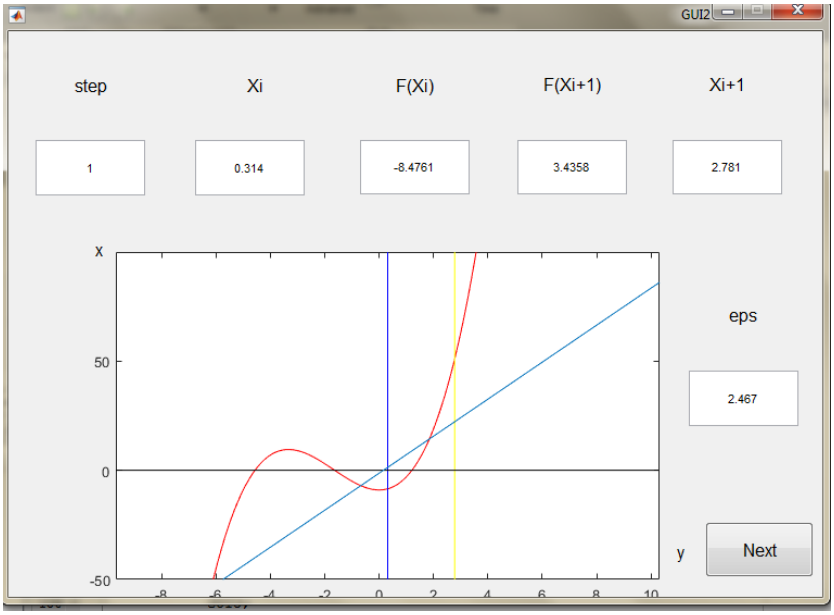
SAMPLE RUNS AND SNAPSHOTS

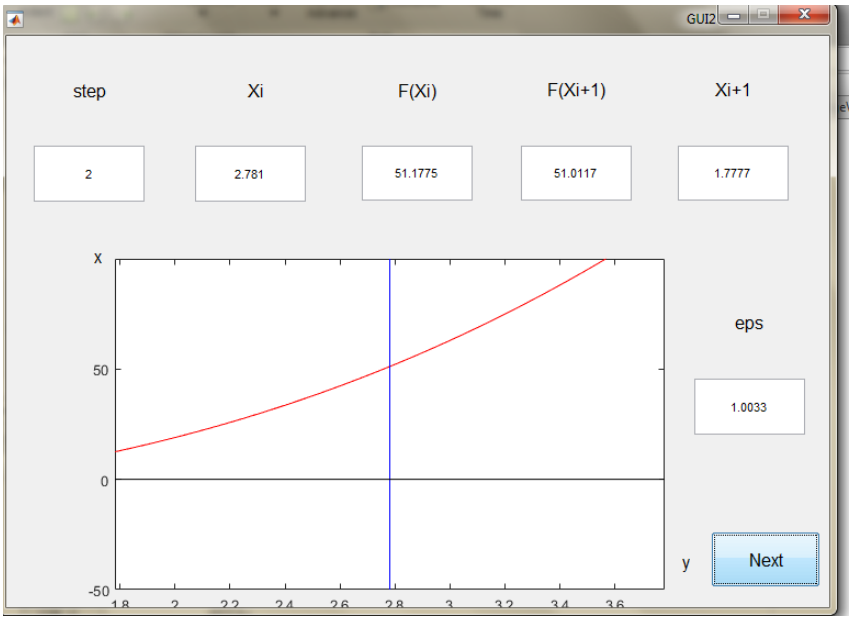
1)Reading from a file:

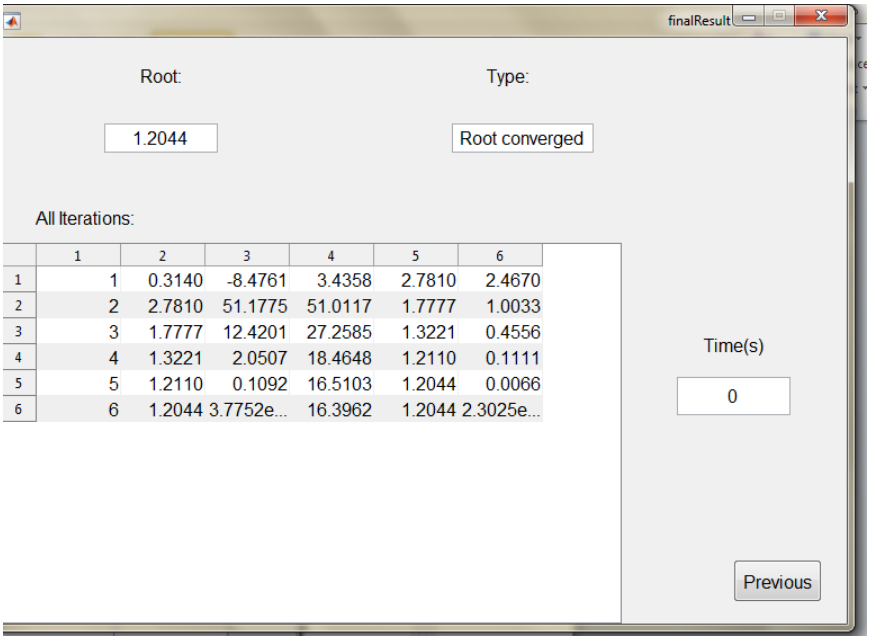




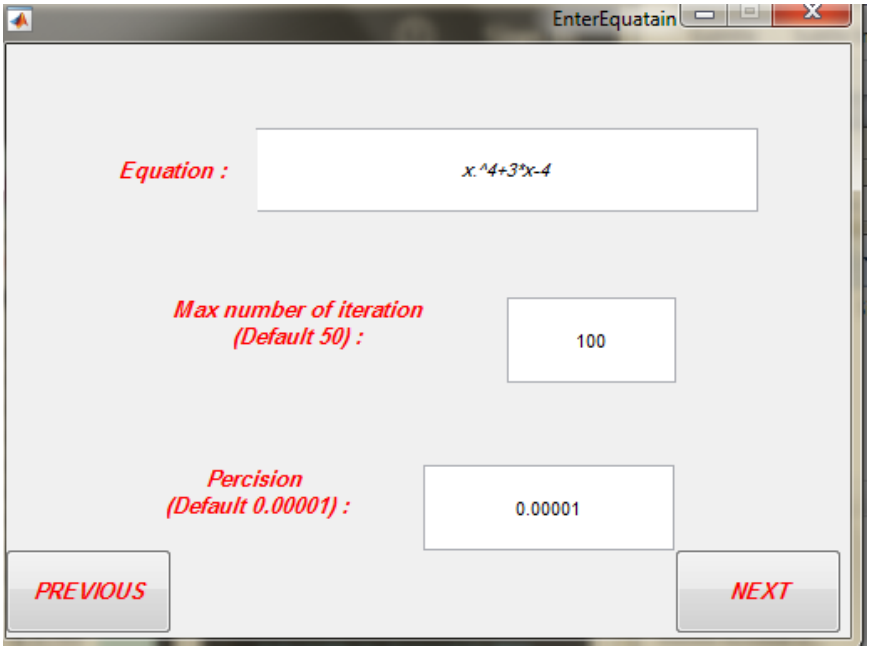
```
x.^3+5*x.^2-9
Newton-Raphson
0.314
0.0001
```

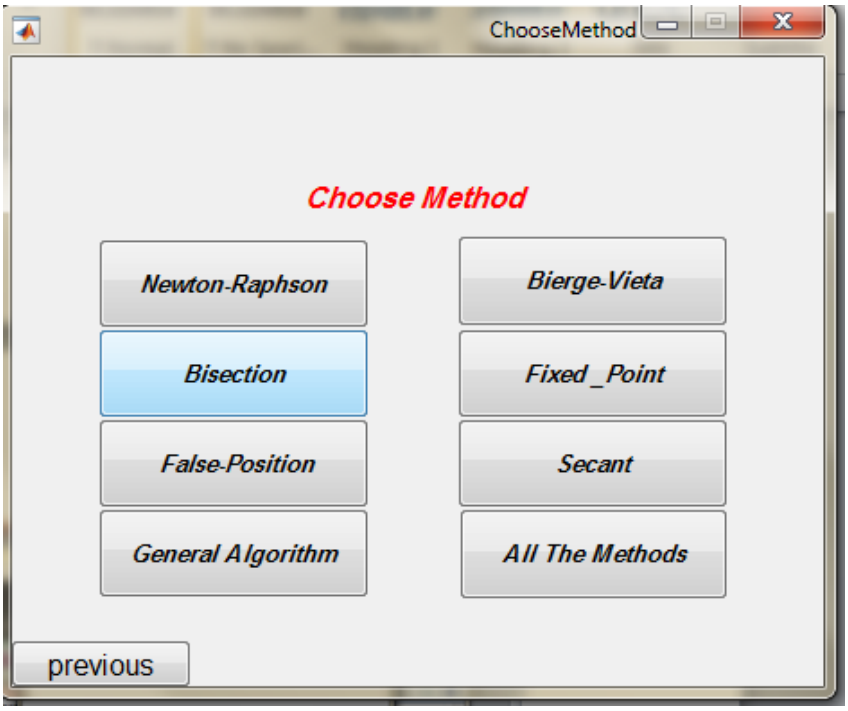


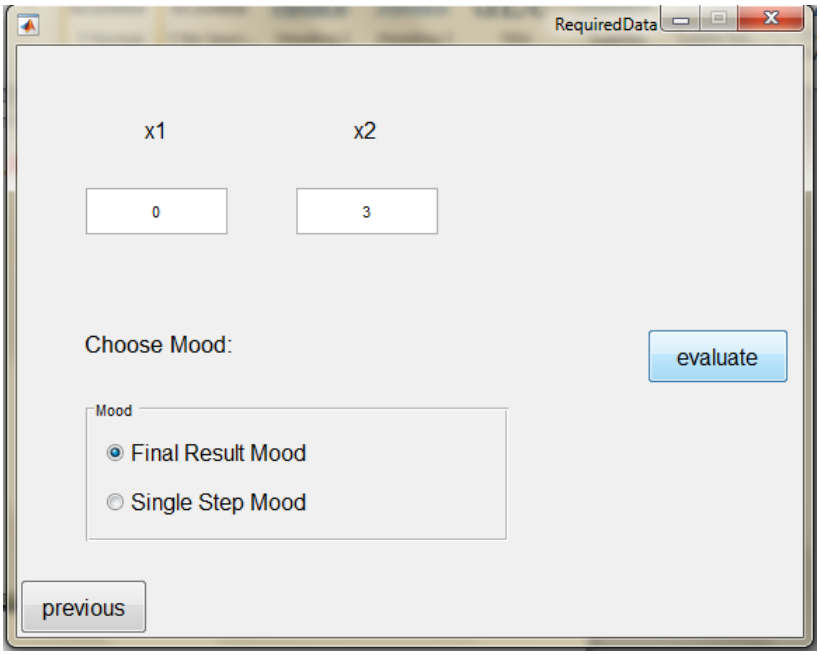


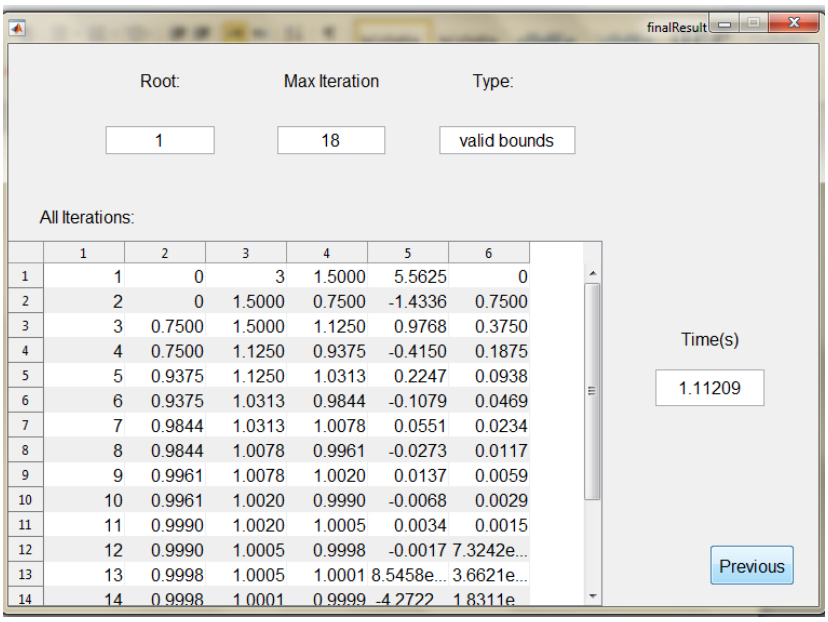


2)Enter an equation:

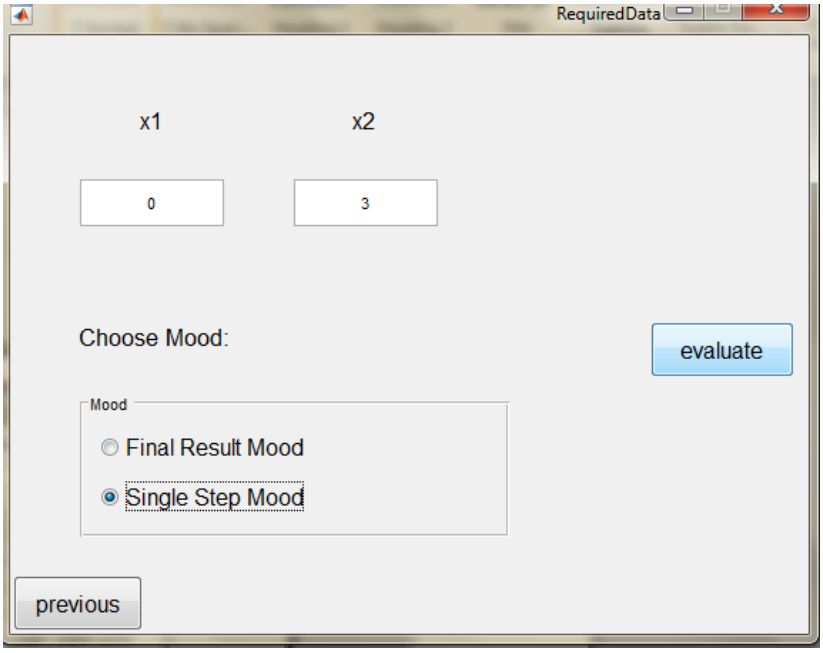


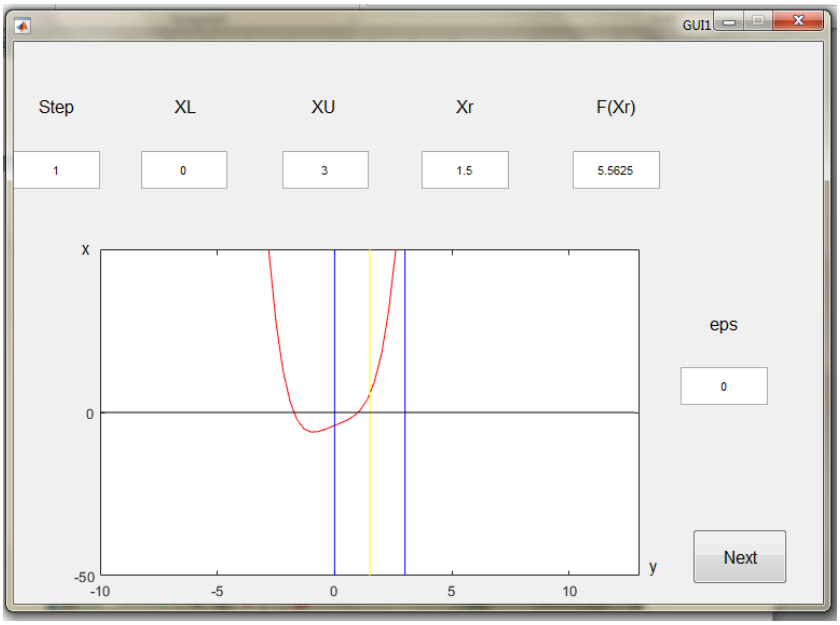

ChooseMethod
Choose Method
Newton-Raphson
Bierge-Vieta
Bisection
Fixed _Point
False-Position
Secant
General Algorithm
All The Methods
previous


RequiredData
x1
x2
0
3
Choose Mood:
evaluate
Mood
Final Result Mood
Single Step Mood
previous

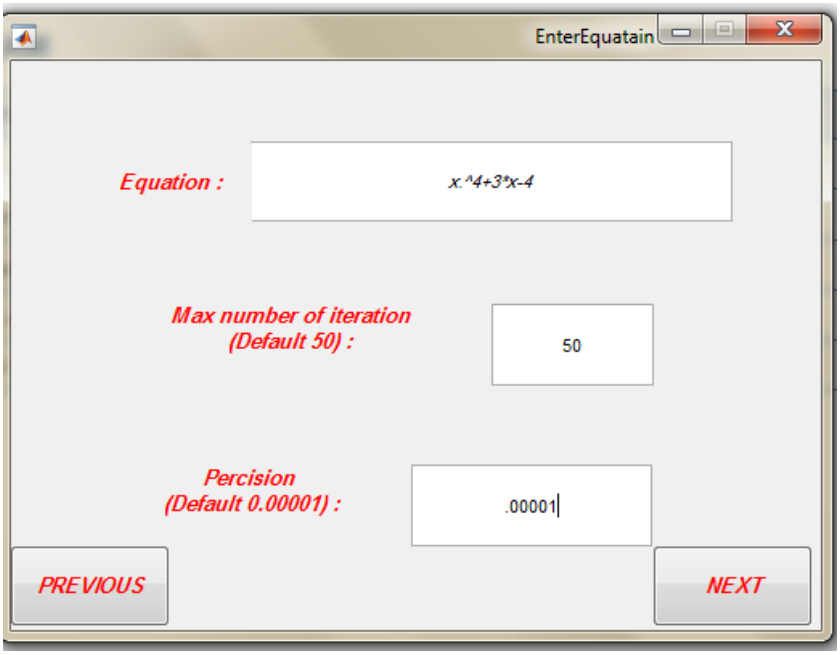


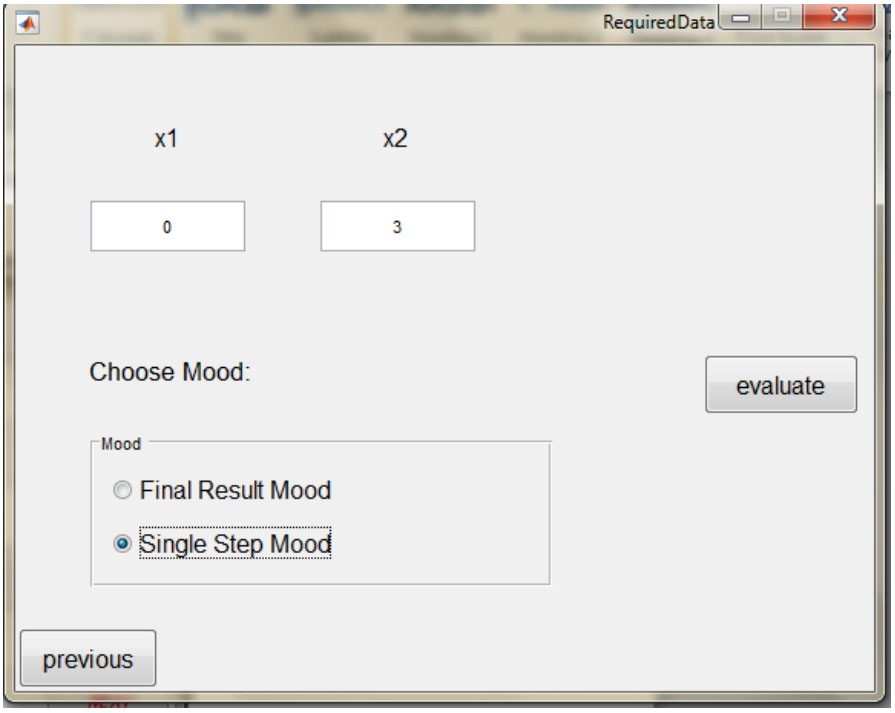
	1	2	3	4	5	6
1	1	0	3	1.5000	5.5625	0
2	2	0	1.5000	0.7500	-1.4336	0.7500
3	3	0.7500	1.5000	1.1250	0.9768	0.3750
4	4	0.7500	1.1250	0.9375	-0.4150	0.1875
5	5	0.9375	1.1250	1.0313	0.2247	0.0938
6	6	0.9375	1.0313	0.9844	-0.1079	0.0469
7	7	0.9844	1.0313	1.0078	0.0551	0.0234
8	8	0.9844	1.0078	0.9961	-0.0273	0.0117
9	9	0.9961	1.0078	1.0020	0.0137	0.0059
10	10	0.9961	1.0020	0.9990	-0.0068	0.0029
11	11	0.9990	1.0020	1.0005	0.0034	0.0015
12	12	0.9990	1.0005	0.9998	-0.0017	7.3242e...
13	13	0.9998	1.0005	1.0001	8.5458e...	3.6621e...
14	14	0.9998	1.0001	0.9999	-4.2722	1.8311e...

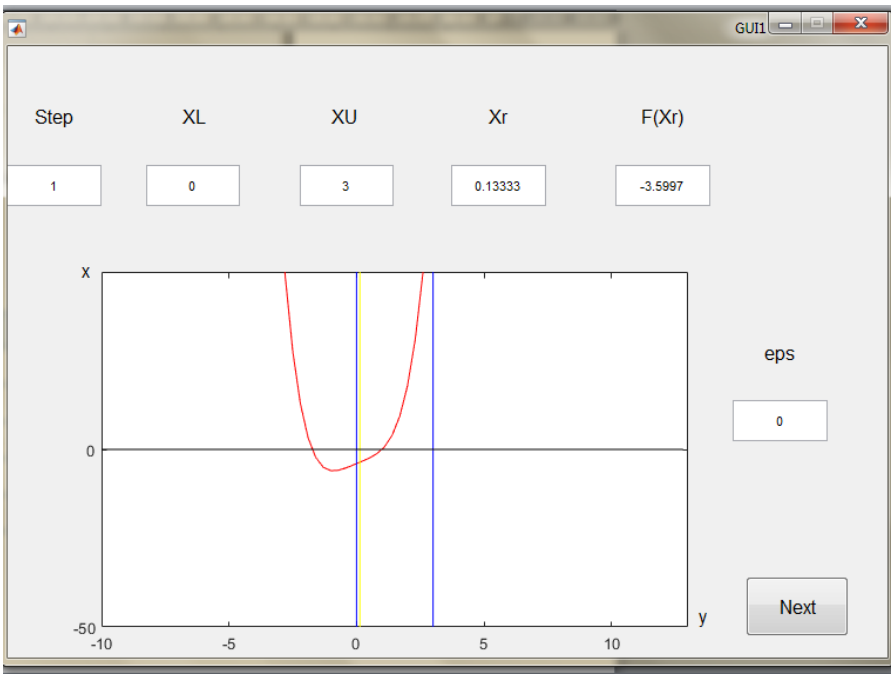


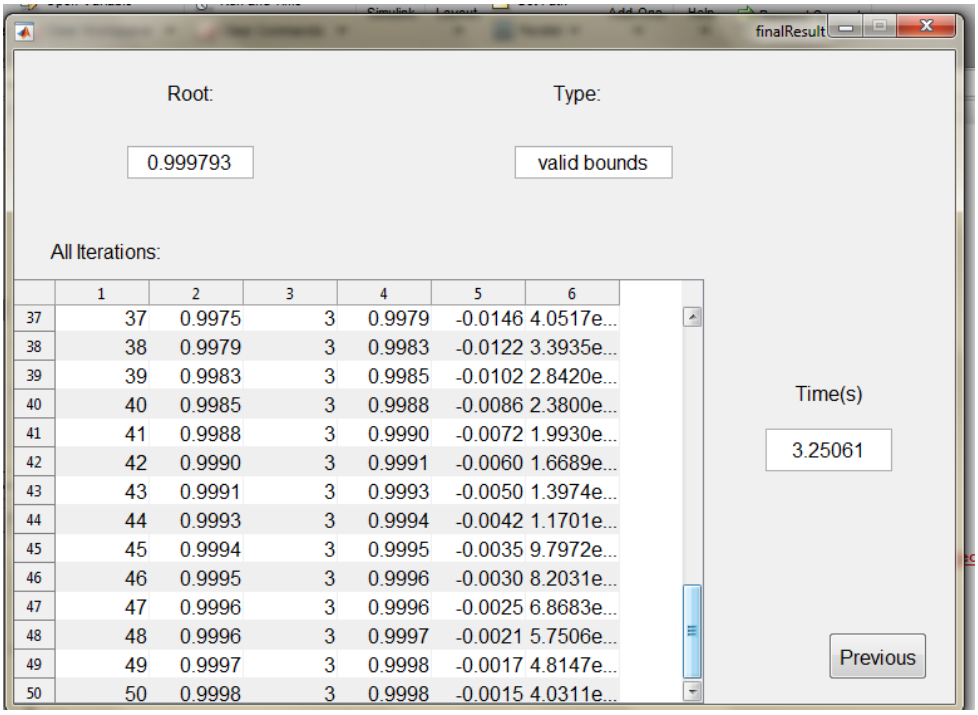


3)False-position example:



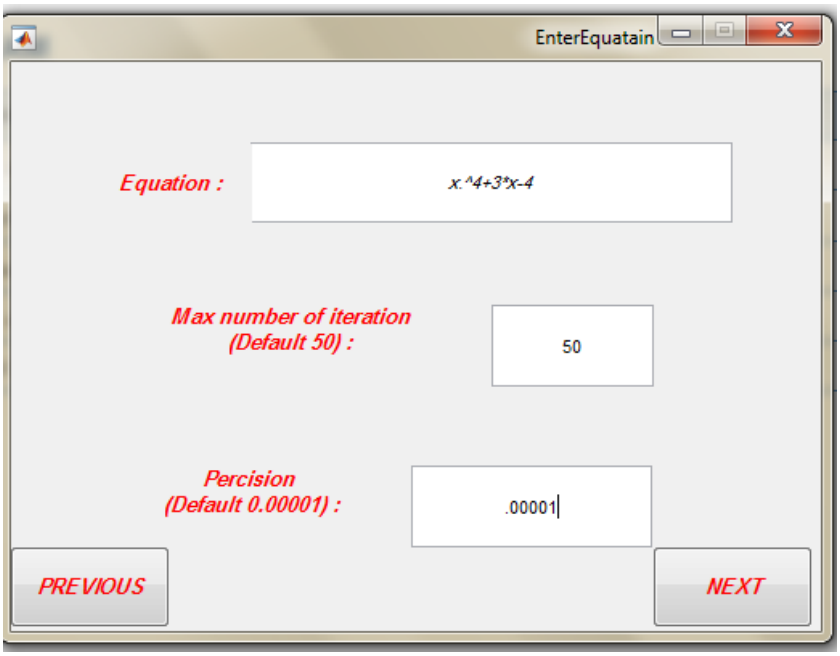

RequiredData
x1
x2
0
3
Choose Mood:
evaluate
Mood
Final Result Mood
Single Step Mood
previous

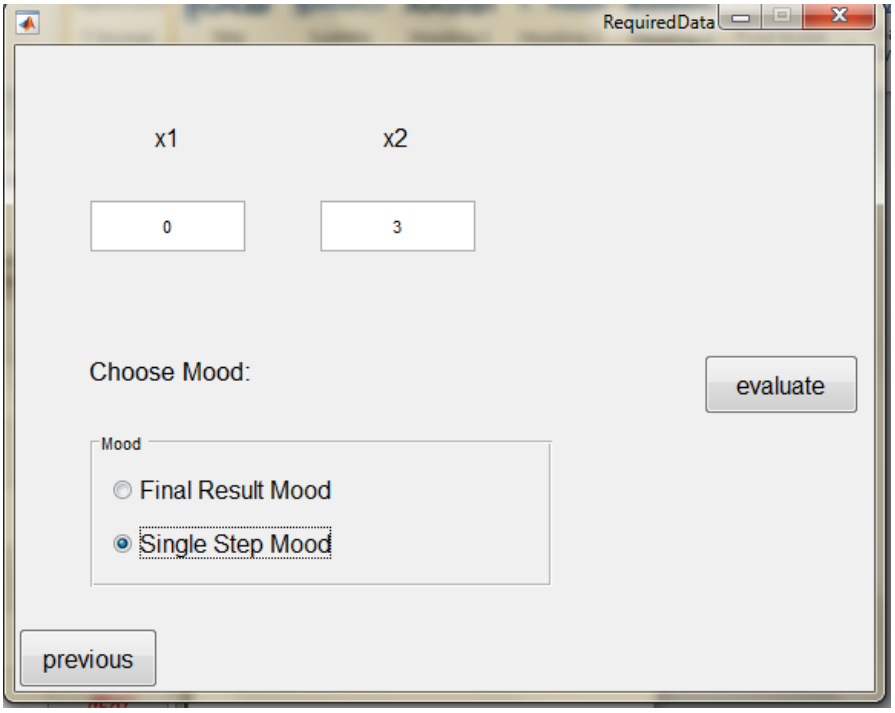

GUI1
Step
XL
XU
Xr
F(Xr)
1
0
3
0.13333
-3.5997
x
0
-50
-10
-5
0
5
10
y
eps
0
Next

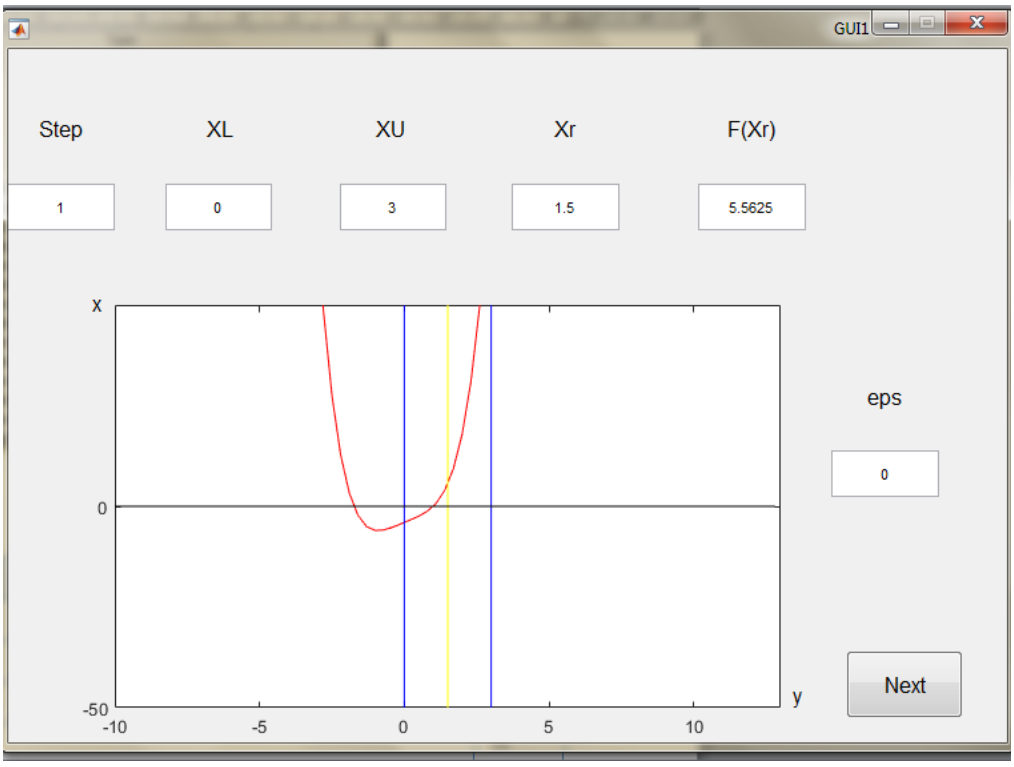


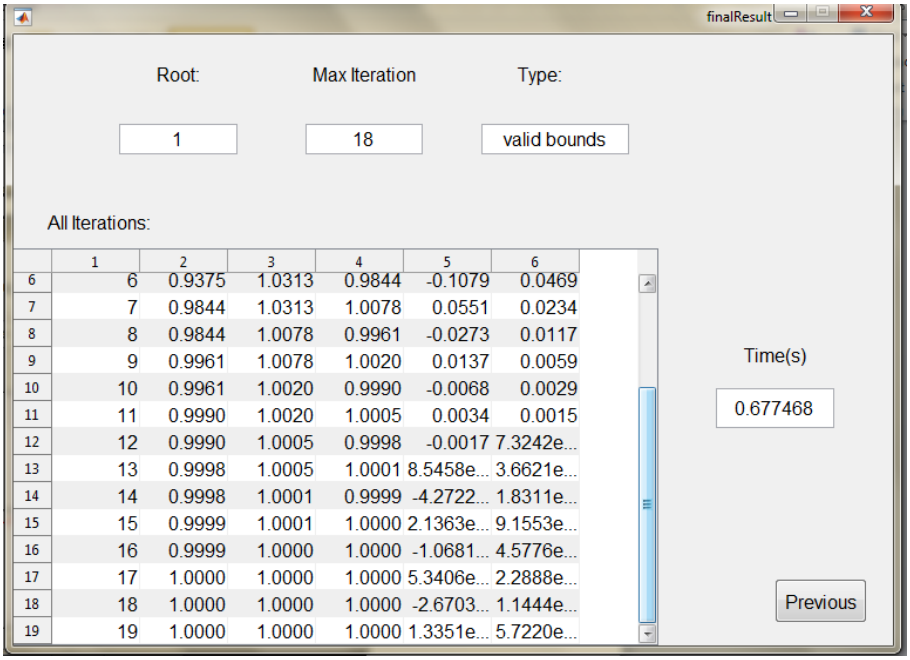
	1	2	3	4	5	6
37	37	0.9975	3	0.9979	-0.0146	4.0517e...
38	38	0.9979	3	0.9983	-0.0122	3.3935e...
39	39	0.9983	3	0.9985	-0.0102	2.8420e...
40	40	0.9985	3	0.9988	-0.0086	2.3800e...
41	41	0.9988	3	0.9990	-0.0072	1.9930e...
42	42	0.9990	3	0.9991	-0.0060	1.6689e...
43	43	0.9991	3	0.9993	-0.0050	1.3974e...
44	44	0.9993	3	0.9994	-0.0042	1.1701e...
45	45	0.9994	3	0.9995	-0.0035	9.7972e...
46	46	0.9995	3	0.9996	-0.0030	8.2031e...
47	47	0.9996	3	0.9996	-0.0025	6.8683e...
48	48	0.9996	3	0.9997	-0.0021	5.7506e...
49	49	0.9997	3	0.9998	-0.0017	4.8147e...
50	50	0.9998	3	0.9998	-0.0015	4.0311e...

2)Bisection example:




RequiredData
x1
x2
0
3
Choose Mood:
evaluate
Mood
Final Result Mood
Single Step Mood
previous

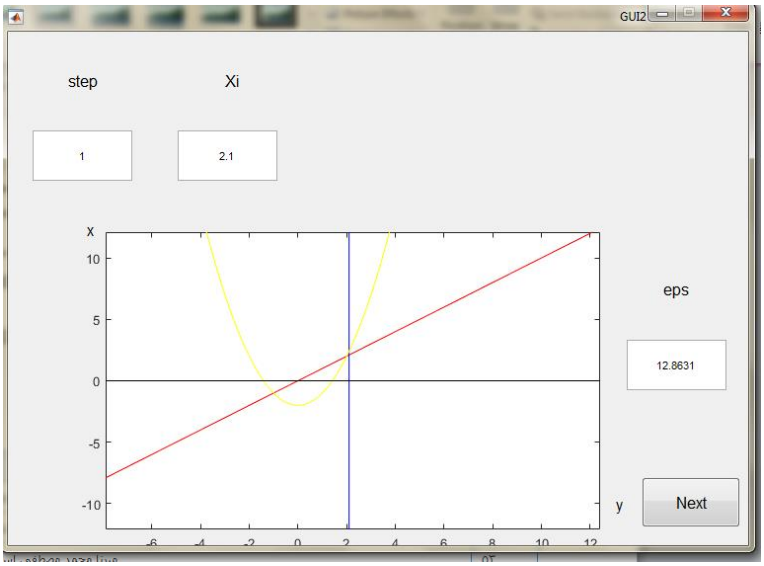

GUI1
Step
XL
XU
Xr
F(Xr)
1
0
3
1.5
5.5625
x
0
-50
-10
-5
0
5
10
y
eps
0
Next



	1	2	3	4	5	6
6	6	0.9375	1.0313	0.9844	-0.1079	0.0469
7	7	0.9844	1.0313	1.0078	0.0551	0.0234
8	8	0.9844	1.0078	0.9961	-0.0273	0.0117
9	9	0.9961	1.0078	1.0020	0.0137	0.0059
10	10	0.9961	1.0020	0.9990	-0.0068	0.0029
11	11	0.9990	1.0020	1.0005	0.0034	0.0015
12	12	0.9990	1.0005	0.9998	-0.0017	7.3242e...
13	13	0.9998	1.0005	1.0001	8.5458e...	3.6621e...
14	14	0.9998	1.0001	0.9999	-4.2722...	1.8311e...
15	15	0.9999	1.0001	1.0000	2.1363e...	9.1553e...
16	16	0.9999	1.0000	1.0000	-1.0681...	4.5776e...
17	17	1.0000	1.0000	1.0000	5.3406e...	2.2888e...
18	18	1.0000	1.0000	1.0000	-2.6703...	1.1444e...
19	19	1.0000	1.0000	1.0000	1.3351e...	5.7220e...

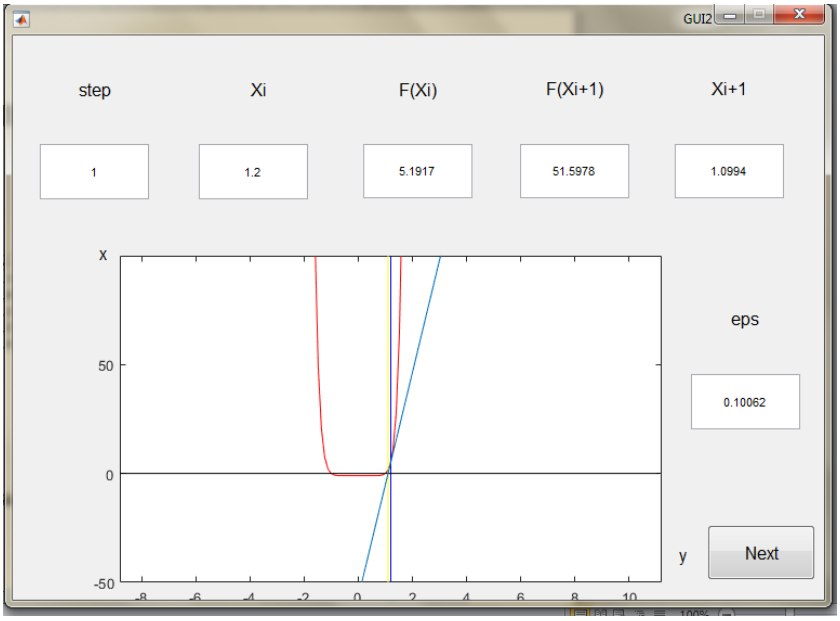
3)fixed point example:

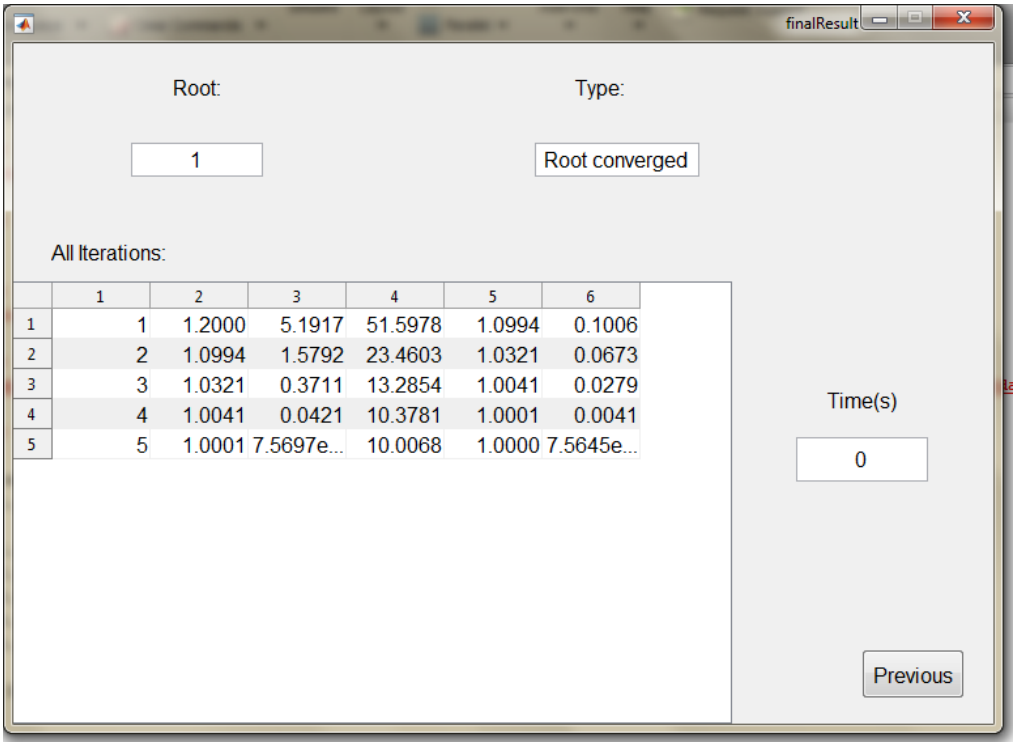
```
x.^2-x-2
Fixed point
2.1
0.0001
```



4)Newton example:

```
x.^10-1
Newton-Raphson
1.2
0.0001
```

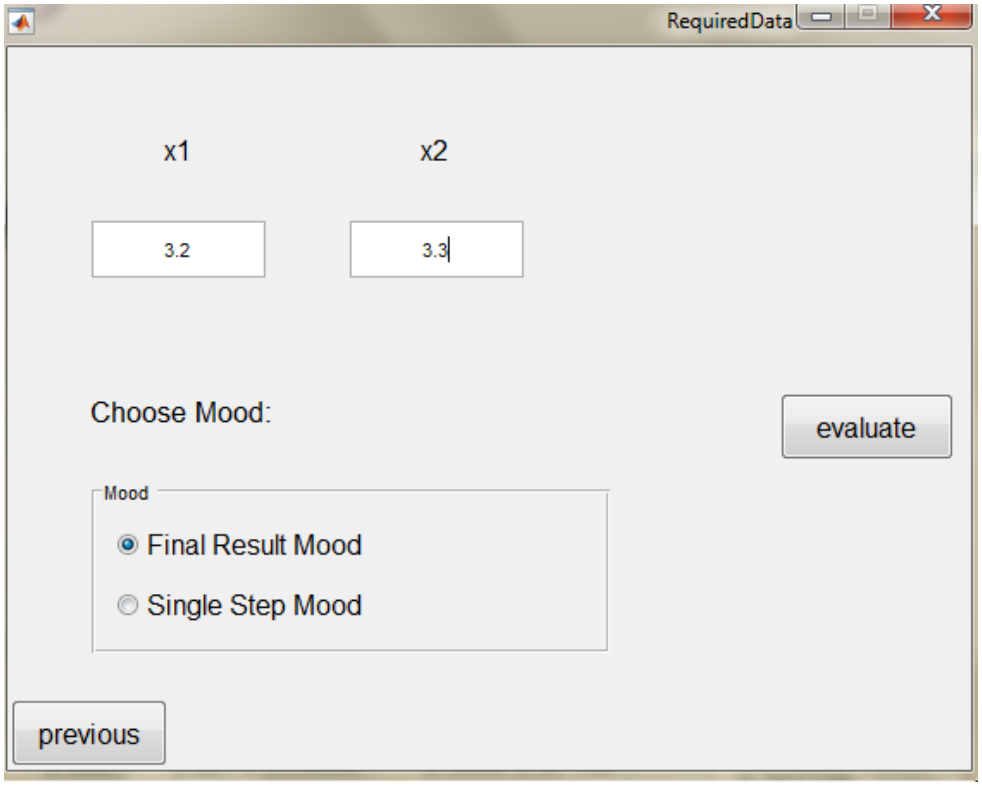


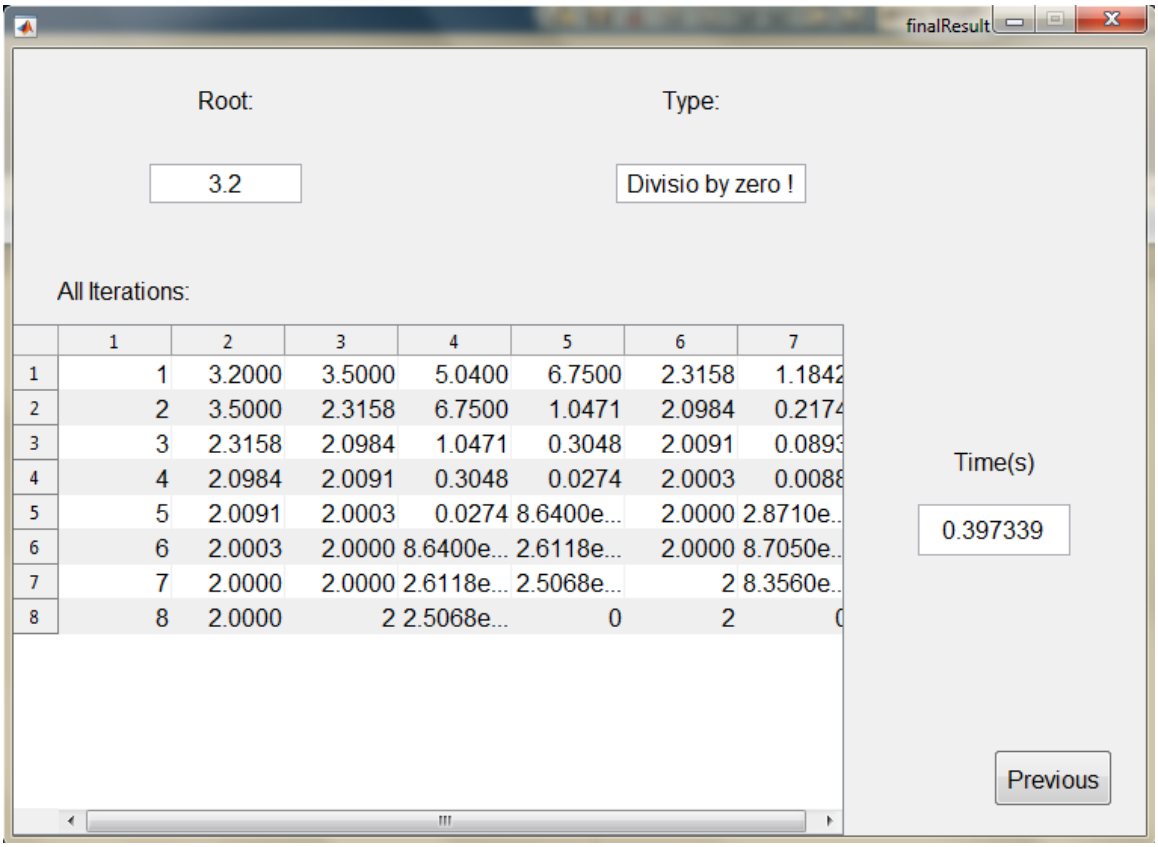


	1	2	3	4	5	6
1	1	1.2000	5.1917	51.5978	1.0994	0.1006
2	2	1.0994	1.5792	23.4603	1.0321	0.0673
3	3	1.0321	0.3711	13.2854	1.0041	0.0279
4	4	1.0041	0.0421	10.3781	1.0001	0.0041
5	5	1.0001	7.5697e...	10.0068	1.0000	7.5645e...

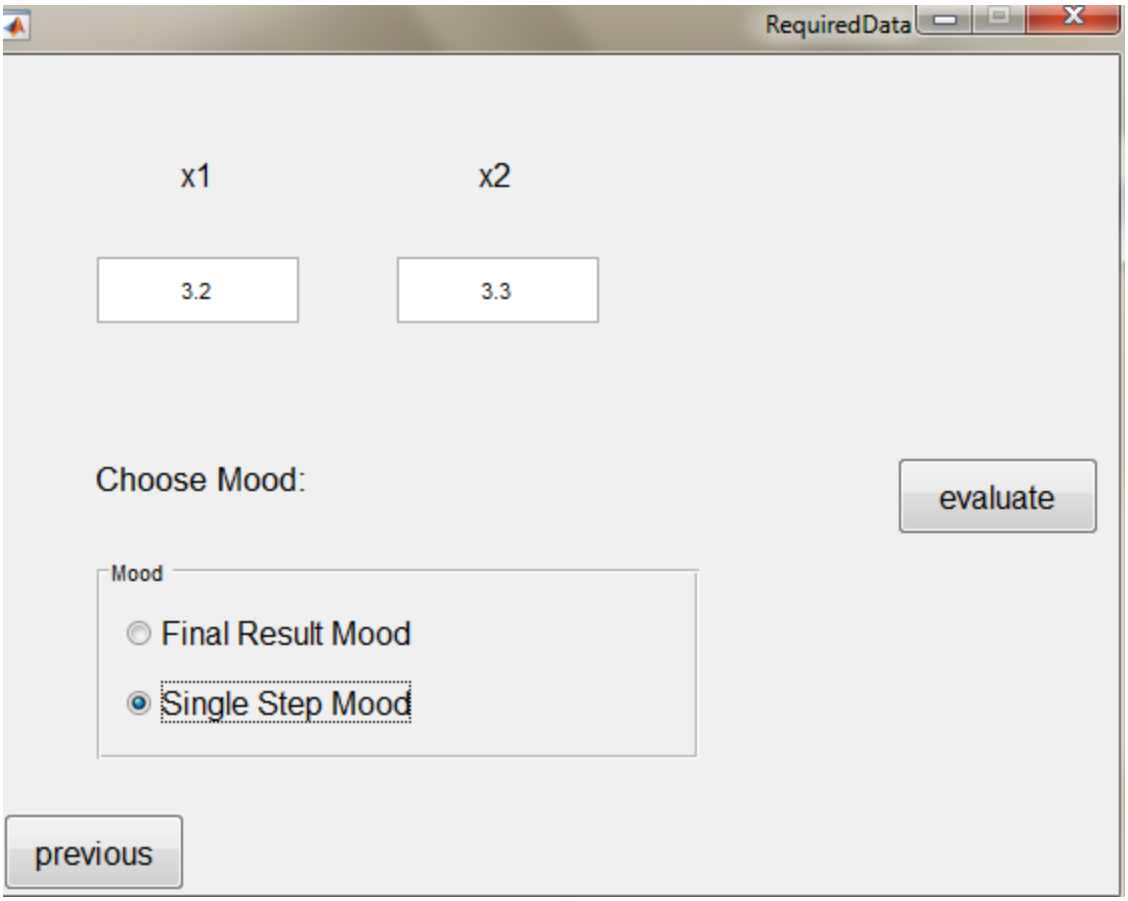
4)Secant example:

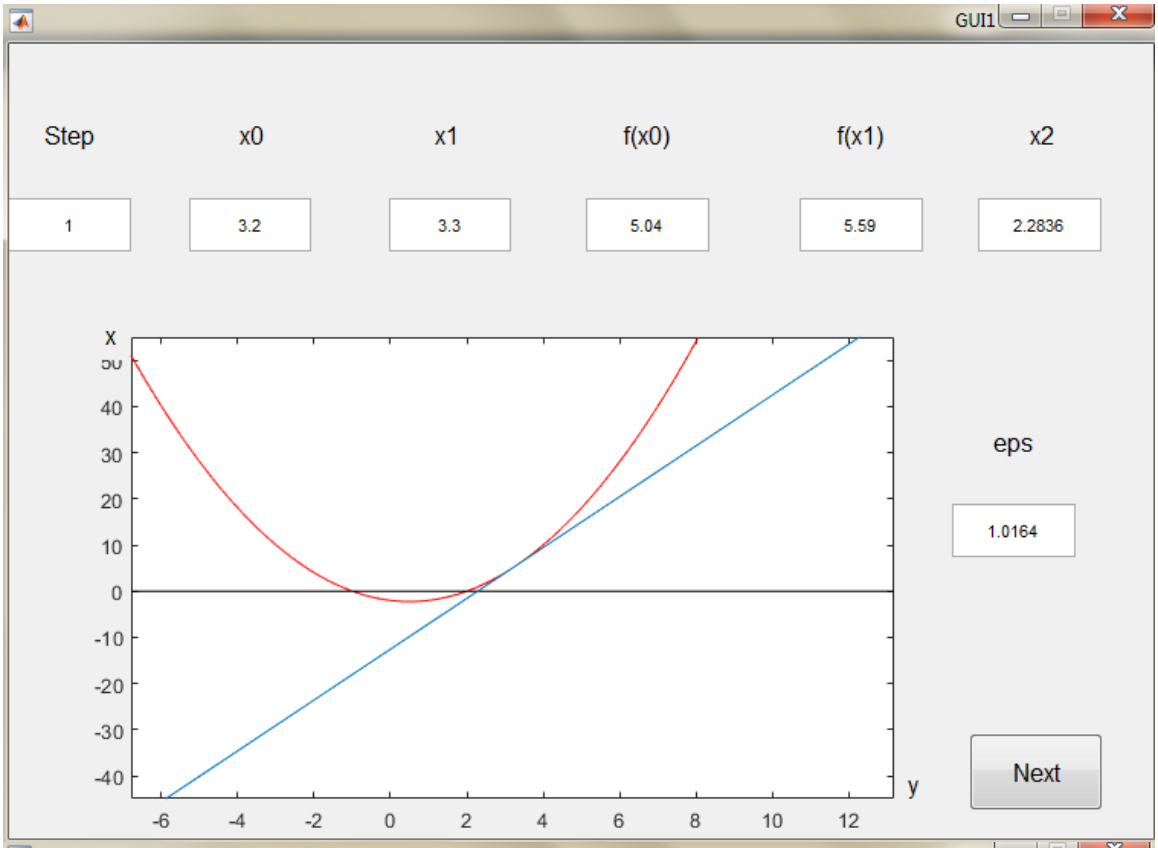
```
x.^2-2
Secant
-3 3
0.0001
```

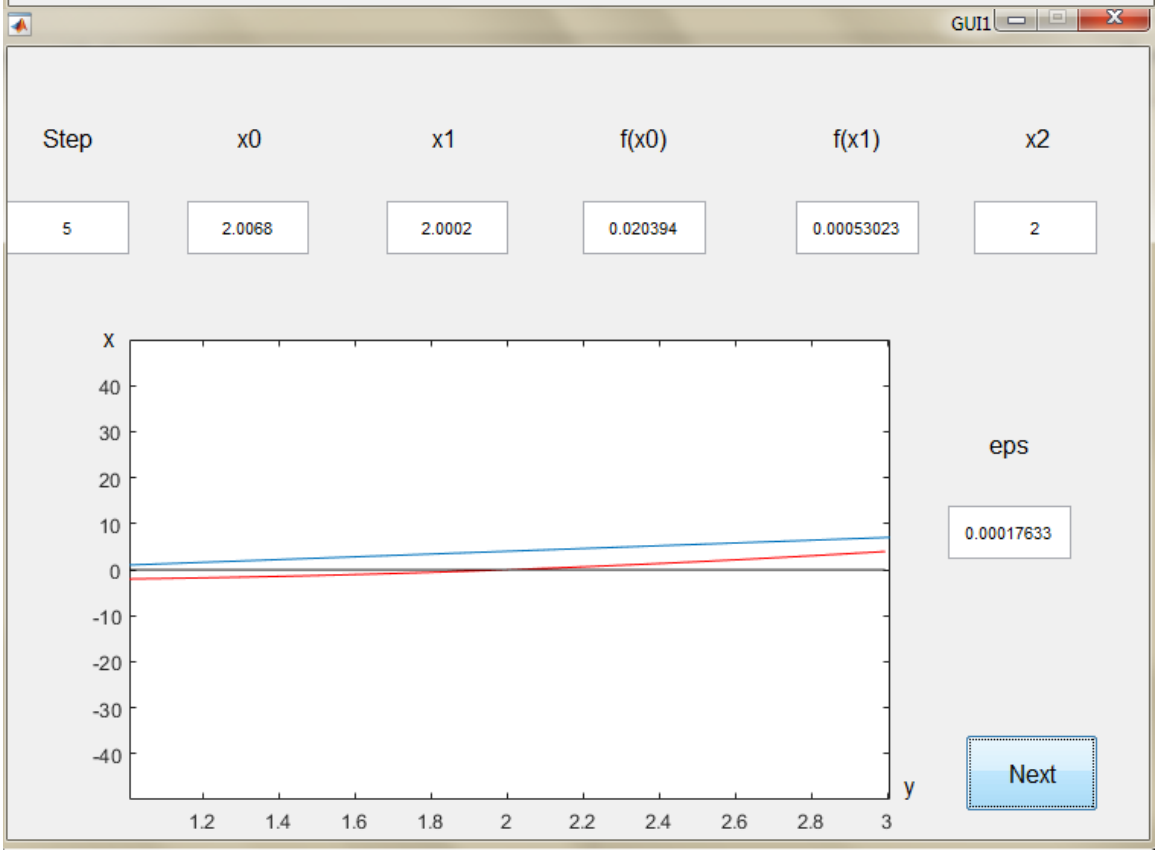




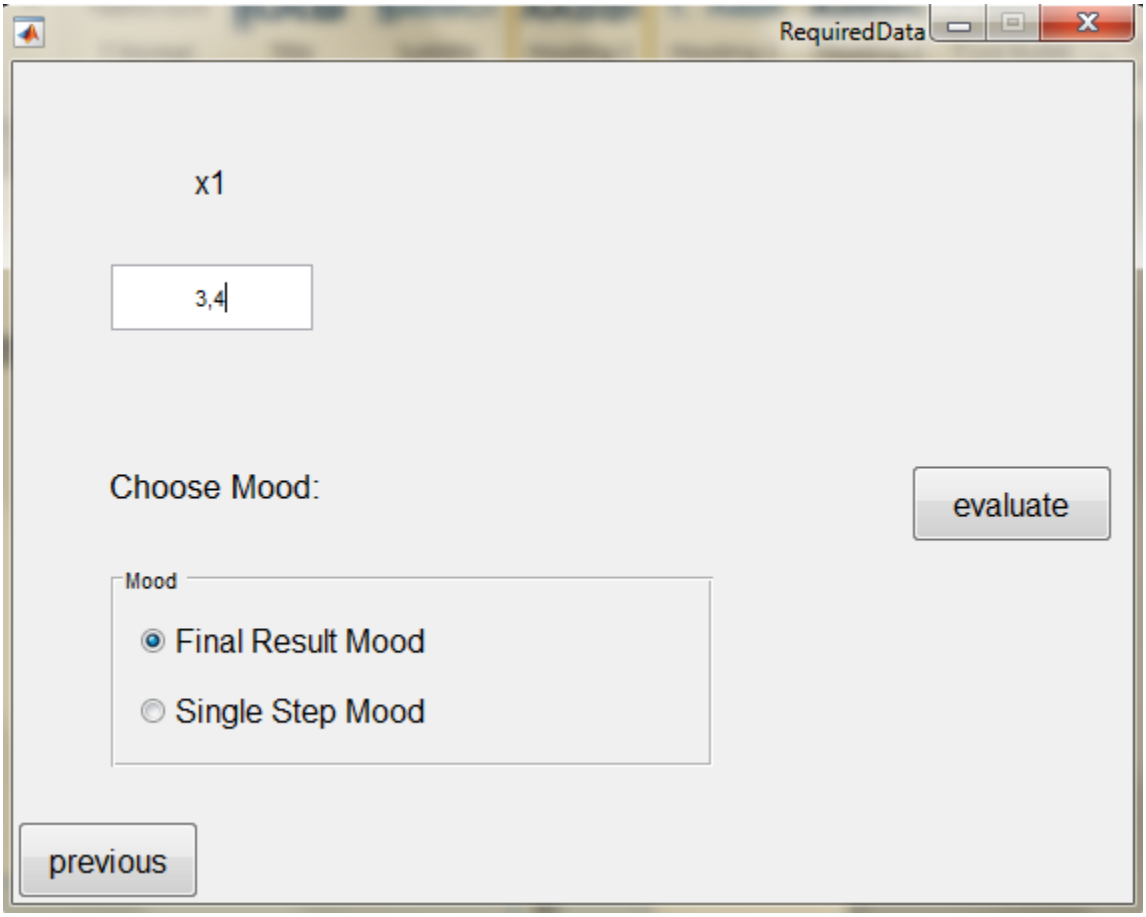
	1	2	3	4	5	6	7
1	1	3.2000	3.5000	5.0400	6.7500	2.3158	1.1842
2	2	3.5000	2.3158	6.7500	1.0471	2.0984	0.2174
3	3	2.3158	2.0984	1.0471	0.3048	2.0091	0.0893
4	4	2.0984	2.0091	0.3048	0.0274	2.0003	0.0088
5	5	2.0091	2.0003	0.0274	8.6400e...	2.0000	2.8710e..
6	6	2.0003	2.0000	8.6400e...	2.6118e...	2.0000	8.7050e..
7	7	2.0000	2.0000	2.6118e...	2.5068e...	2	8.3560e..
8	8	2.0000	2	2.5068e...	0	2	0

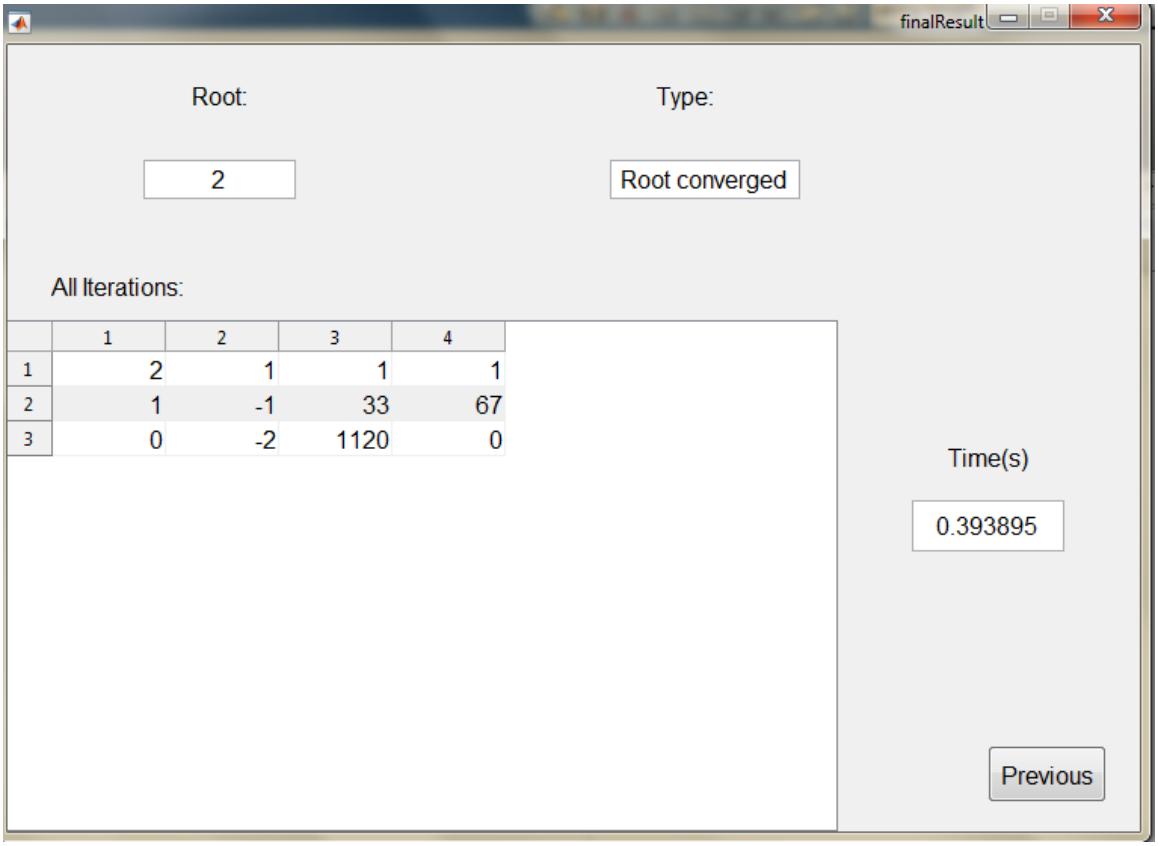

RequiredData
x1
x2
3.2
3.3
Choose Mood:
evaluate
Mood
Final Result Mood
Single Step Mood
previous

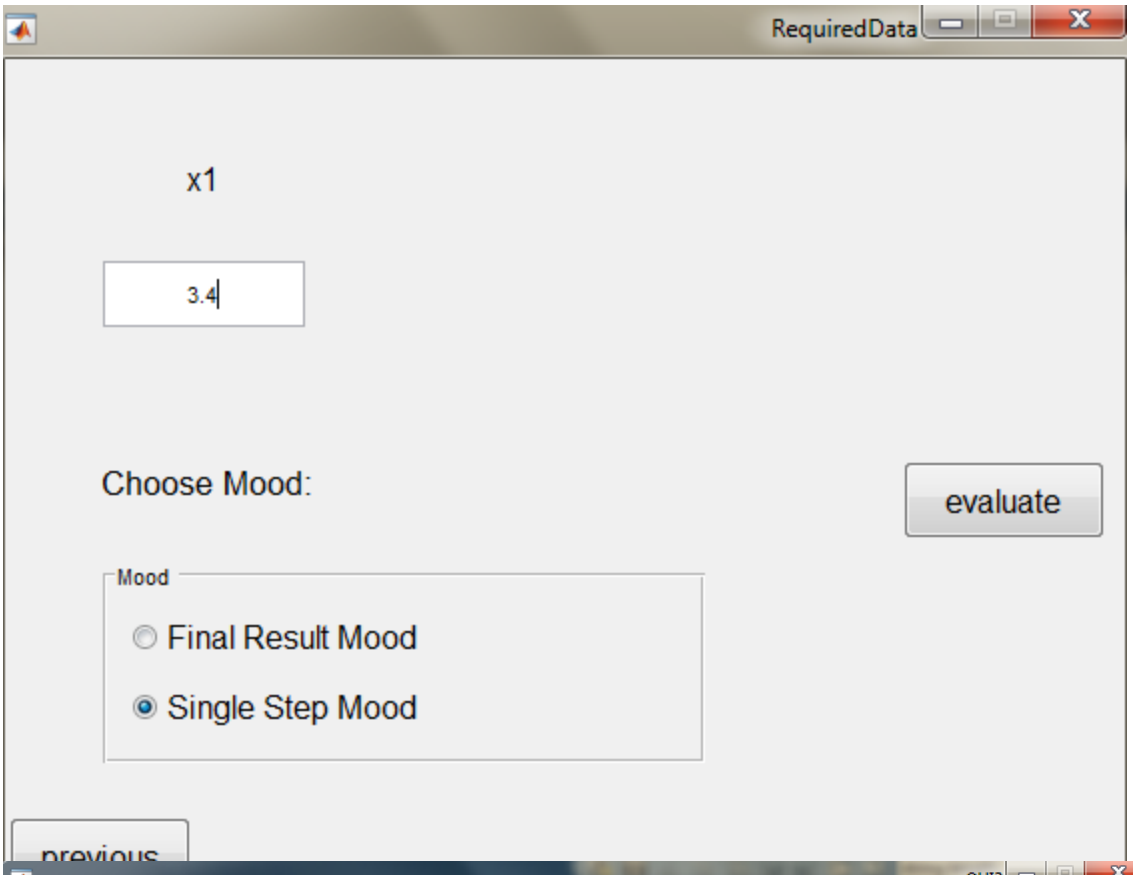

GUI1
Step
x0
x1
f(x0)
f(x1)
x2
1
3.2
3.3
5.04
5.59
2.2836
x
y
eps
1.0164
Next

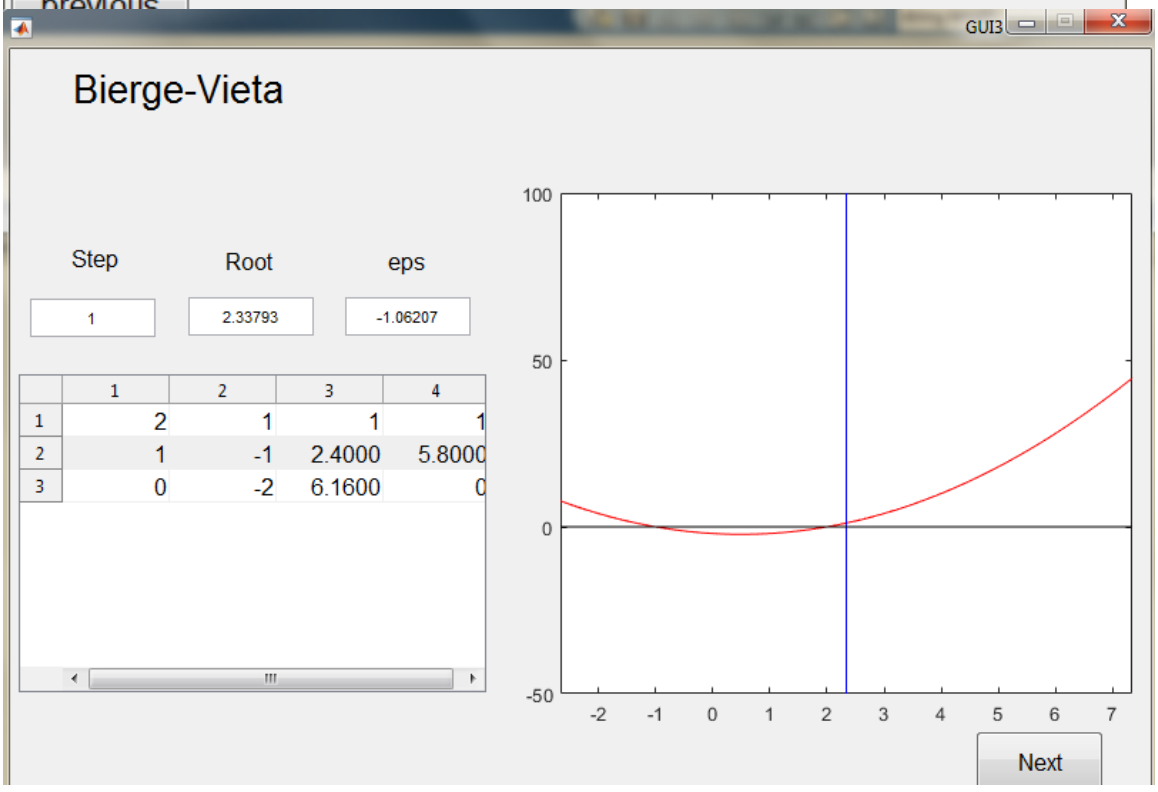

GUI1
Step
x0
x1
f(x0)
f(x1)
x2
5
2.0068
2.0002
0.020394
0.00053023
2
x
y
eps
0.00017633
Next

Beirge vieta

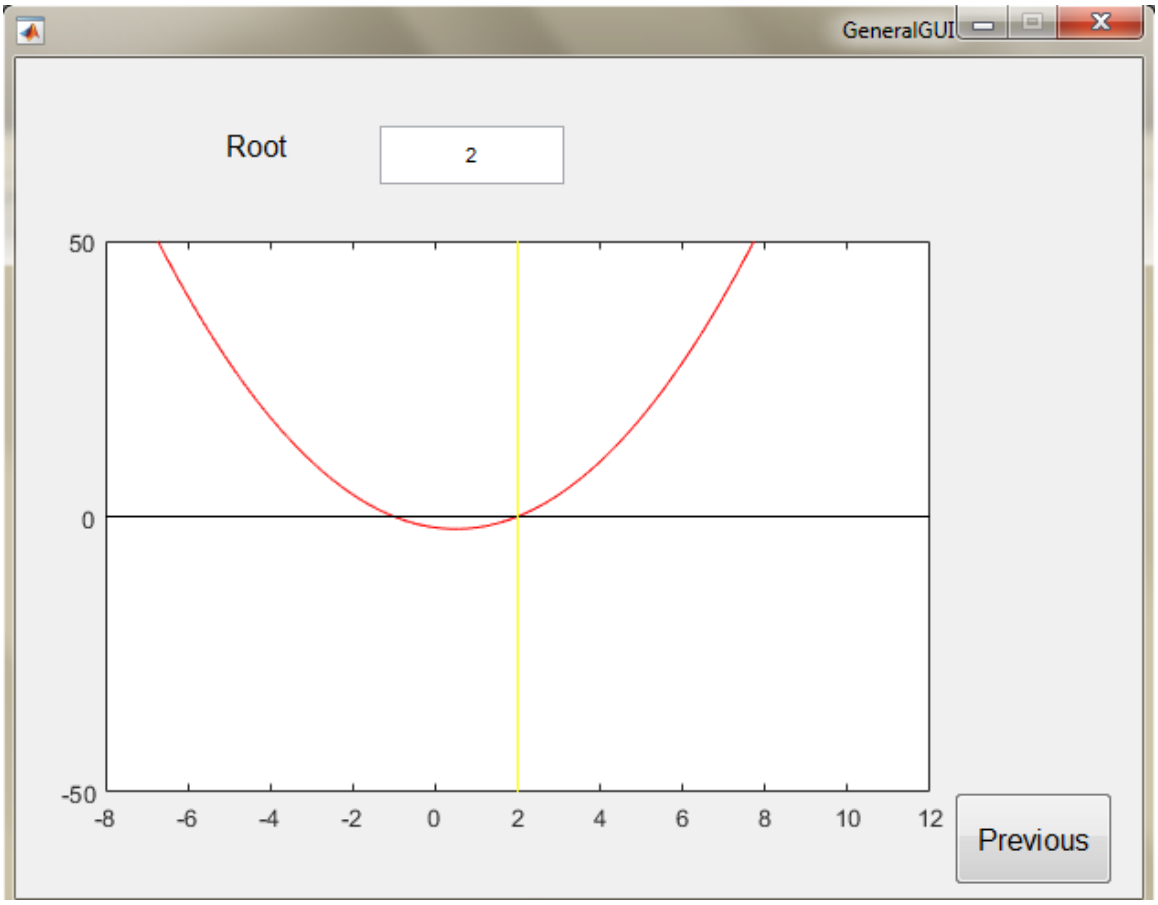



finalResult
Root:
Type:
2
Root converged
All Iterations:
1 2 3 4
1 2 1 1 1
2 1 -1 33 67
3 0 -2 1120 0
Time(s)
0.393895
Previous


RequiredData
x1
3.4
Choose Mood:
evaluate
Mood
Final Result Mood
Single Step Mood
previous


GUI3
Bierge-Vieta
Step
Root
eps
1
2.33793
-1.06207
1 2 3 4
1 2 1 1 1
2 1 -1 2.4000 5.8000
3 0 -2 6.1600 0
100
50
0
-50
-2 -1 0 1 2 3 4 5 6 7
Next

General Algorithm



TEAM

Name	ID
ايمان رفيق عبد القادر محمد على	11
تقى علاء احمد الجندى	14
ميرنا محمد مصطفى اسماعيل مصطفى	٥٣
ندى سلامه محمد على	٥٥
يمنى جمال الدين محمود السيد	60