Numerical Analysis

Project 1

Final Report

Abdalla Gamal Mubarak 6771

Gui functions:

Create figures for building apps using uifigure.

Create a table user interface component using uitable.

msgbox: Generate a message box.

Create a dialogue box called inputdlg to collect user input.

uiwait: Stop running programmes and wait for them to start again.

Convert integers to a character array using num2str.

Convert strings to double precision values using str2double.

Functions used:

Fzero: The root of a nonlinear function. Fevel: Evaluates a function using its name or its handle and the input arguments x1...xM.

Data Structures:

1-Toc:

Toc reads the amount of time that has passed since the tic function's call to start the stopwatch timer. When the toc function is executed, MATLAB® reads the internal time and shows the amount of time that has passed since the most recent call to the tic function without producing anything. The amount of time elapsed is shown in seconds. The time since the call to the tic function corresponding to timerVal is shown by the function toc(timerVal).

2-Tic:

Tic uses the toc function to calculate the passage of time. The toc function takes the recorded value from the tic function to compute the amount of time that has passed since the current time was recorded. In order to explicitly send the current time to the toc method, timerVal = tic puts the time in timerVal. When there are many calls to tic to time various sections of the same code, passing this value is helpful. An integer called timerVal only has significance for the toc function.

3-HPF decimal class:

I frequently observe folks requesting a tool with an accuracy range of greater than 16 digits. Numerical analysis guiding principles are significantly more valuable than any high precision instrument. However, there are situations when having a little additional accuracy is useful. Some of you will just want to play in the sandbox of enormous numbers. While some of you might make use of

Ben Barrowes' tools, HPF is entirely built in MATLAB, so compiles are not necessary. Whatever your motivations, I offer HPF, a high precision floating point tool, to all of you.

4- B=ArrayFun(func,A):

One by one, apply the function func to the components of A. The outputs from func are then concatenated by arrayfun into the output array B, making B(i) equal to func(A(i)) for the ith element of A. A function handle to a function that accepts one input parameter and returns a scalar is provided as the input argument func. As long as objects of that type can be concatenated, any data type can be used for func's output. The dimensions of arrays A and B are the same.

1)Bisection

The algorithm we used:

1. Start

2. Define function f(x)

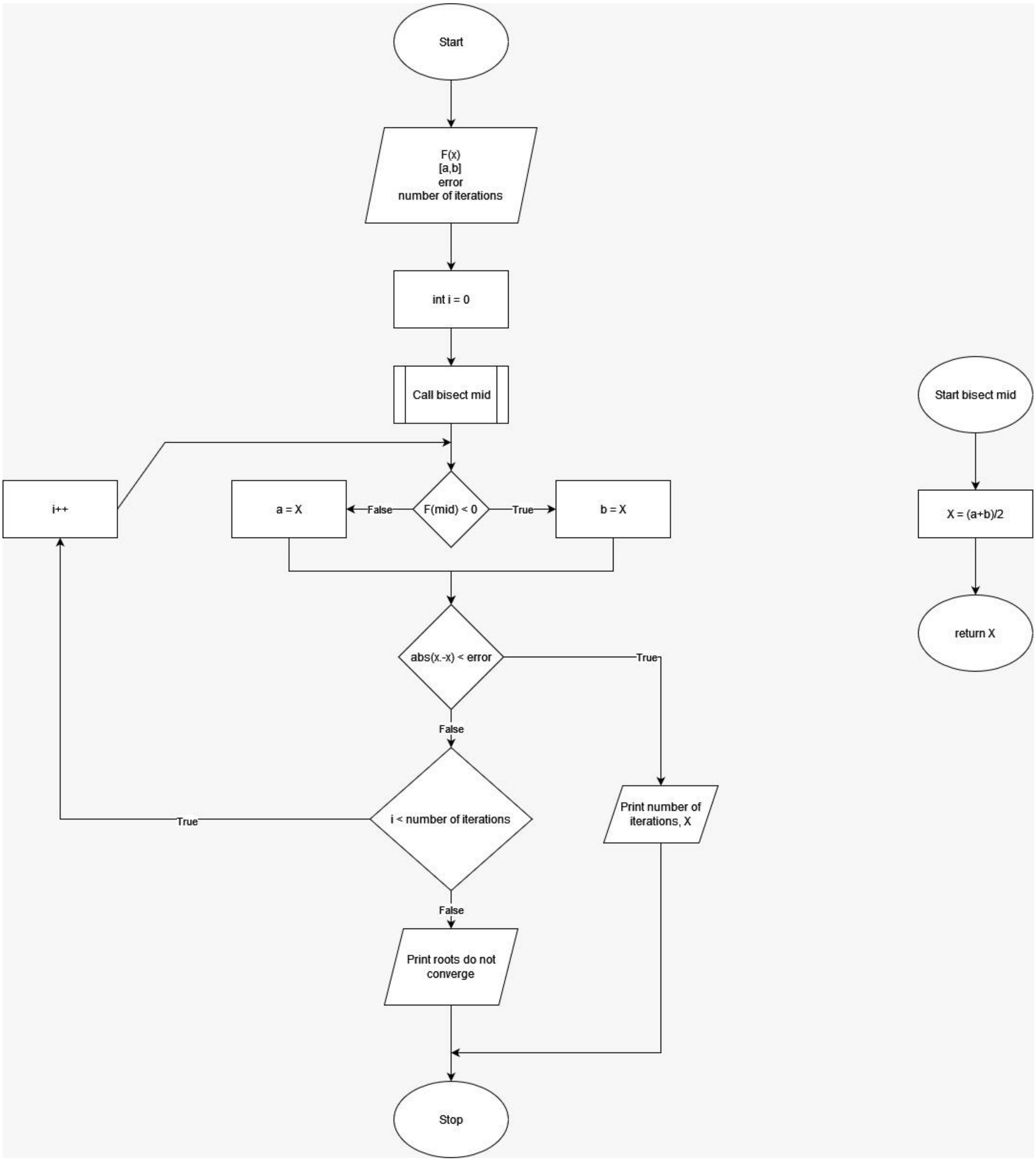
3. Input a. Lower and Upper guesses x0 and x1 b. tolerable error e

4. If f(x0)*f(x1) > 0 print "Incorrect initial guesses" goto 3 End If

5. Do x2 = (x0+x1)/2 If f(x0)*f(x2) < 0 x1 = x2 Else x0 = x2 End If while abs(f(x2) > e

6. Print root as x2

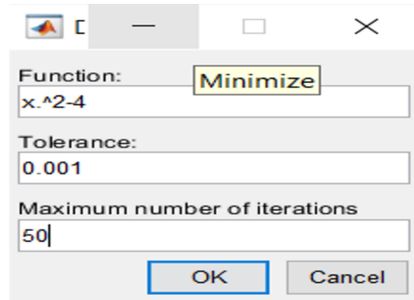
7. Stop


Start
F(x)
[a,b]
error
number of iterations
int i = 0
Call bisect mid
F(mid) < 0
False
True
a = X
b = X
i++
abs(x.-x) < error
True
False
i < number of iterations
True
False
Print number of iterations, X
Print roots do not converge
Stop
Start bisect mid
X = (a+b)/2
return X

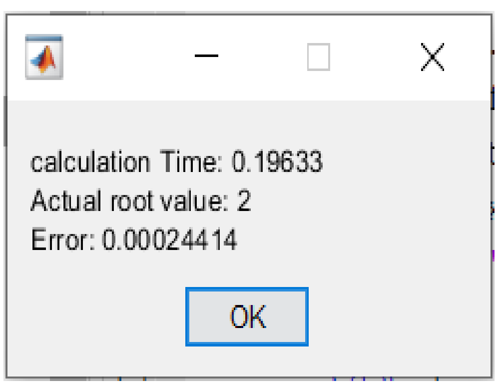
A)f(x) =X.^2-4

Tolerance =0.001

Interval =[0,3]



Step	X-Lower	X-Upper	X-Root	Error %
1	0	3	1.5	-
2	1.5	3	2.25	33.33333333
3	1.5	2.25	1.875	20
4	1.875	2.25	2.0625	9.090909091
5	1.875	2.0625	1.96875	4.761904762
6	1.96875	2.0625	2.015625000	2.325581395
7	1.96875	2.015625000	1.992187500	1.176470588
8	1.992187500	2.015625000	2.003906250	0.5847953216
9	1.992187500	2.003906250	1.998046875	0.2932551320
10	1.998046875	2.003906250	2.000976562	0.1464128843
11	1.998046875	2.000976562	1.999511719	0.07326007326
12	1.999511719	2.000976562	2.000244141	0.03661662395



B) f(x) = 3x^4 + 6.1x^3 - 2x^2 + 3x +2

Tolerance =0.001

Interval = [-1,0]

Step	X-Lower	X-Upper	X-Root	Error %
1	-1	0	-0.5	-
2	-0.5	0	-0.25	100
3	-0.5	-0.25	-0.375	33.33333333
4	-0.5	-0.375	-0.4375	14.28571429
5	-0.4375	-0.375	-0.40625	7.692307692
6	-0.4375	-0.40625	-0.421875	3.703703704
7	-0.4375	-0.421875	-0.4296875	1.818181818
8	-0.4296875	-0.421875	-0.4257812500	0.9174311927
9	-0.4257812500	-0.421875	-0.4238281250	0.4608294931
10	-0.4257812500	-0.4238281250	-0.4248046875	0.2298850575
11	-0.4248046875	-0.4238281250	-0.4243164063	0.1150747986
12	-0.4243164063	-0.4238281250	-0.4240722656	0.05757052389
13	-0.4240722656	-0.4238281250	-0.4239501953	0.02879355024

calculation Time: 0.0623
Actual root value: -0.42406
Error: 0.00011112

OK

C) $f(x) = x^2 - \sin(x) - 0.5$

Tolerance =0.001

Interval = [0, 2]

Step	X-Lower	X-Upper	X-Root	Error %
1	0	2	1	-
2	1	2	1.5	33.33333333
3	1	1.5	1.25	20
4	1	1.25	1.125	11.11111111
5	1.125	1.25	1.1875	5.263157895
6	1.1875	1.25	1.21875	2.564102564
7	1.1875	1.21875	1.203125000	1.298701299
8	1.1875	1.203125000	1.195312500	0.6535947712
9	1.195312500	1.203125000	1.199218750	0.3257328990
10	1.195312500	1.199218750	1.197265625	0.1631321370
11	1.195312500	1.197265625	1.196289063	0.08163265306
12	1.195312500	1.196289063	1.195800781	0.04083299306
13	1.195800781	1.196289063	1.196044922	0.02041232905
14	1.196044922	1.196289063	1.196166992	0.01020512297
15	1.196044922	1.196166992	1.196105957	0.005102821860
16	1.196044922	1.196105957	1.196075439	0.002551476029
17	1.196075439	1.196105957	1.196090698	0.001275721740
18	1.196075439	1.196090698	1.196083069	6.378649385e-4
19	1.196075439	1.196083069	1.196079254	3.189334865e-4

calculation Time: 0.18004
Actual root value: 1.1961
Error: -2.7791e-06

OK

D) f(x)=x3-x-1

Tolerance= 10^-5

Interval= [1,2]

from input file

```
1   x.^3-x-1
```

Step	X-Lower	X-Upper	X-Root	Error %
1	1	2	1.5	-
2	1	1.5	1.25	20
3	1.25	1.5	1.375	9.090909091
4	1.25	1.375	1.3125	4.761904762
5	1.3125	1.375	1.34375	2.325581395
6	1.3125	1.34375	1.328125000	1.176470588
7	1.3125	1.328125000	1.320312500	0.5917159763
8	1.320312500	1.328125000	1.324218750	0.2949852507
9	1.324218750	1.328125000	1.326171875	0.1472754050
10	1.324218750	1.326171875	1.325195313	0.07369196758
11	1.324218750	1.325195312	1.324707031	0.03685956506
12	1.324707031	1.325195312	1.324951172	0.01842638659
13	1.324707031	1.324951172	1.324829102	0.009214042200
14	1.324707031	1.324829102	1.324768066	0.004607233356
15	1.324707031	1.324768066	1.324737549	0.002303669746
16	1.324707031	1.324737549	1.324722290	0.001151848140
17	1.324707031	1.324722290	1.324714661	5.759273871e-4
18	1.324714661	1.324722290	1.324718475	2.879628643e-4

calculation Time: 0.039553
Actual root value: 1.3247
Error: 5.181e-07

OK

2)Regula Falsi:

The algorithm we used:

1. Start

2. Define function f(x)

3. Input a. Lower and Upper guesses x0 and x1 b. tolerable error e

4. If f(x0)*f(x1) > 0

 print "Incorrect initial guesses" goto

 End If

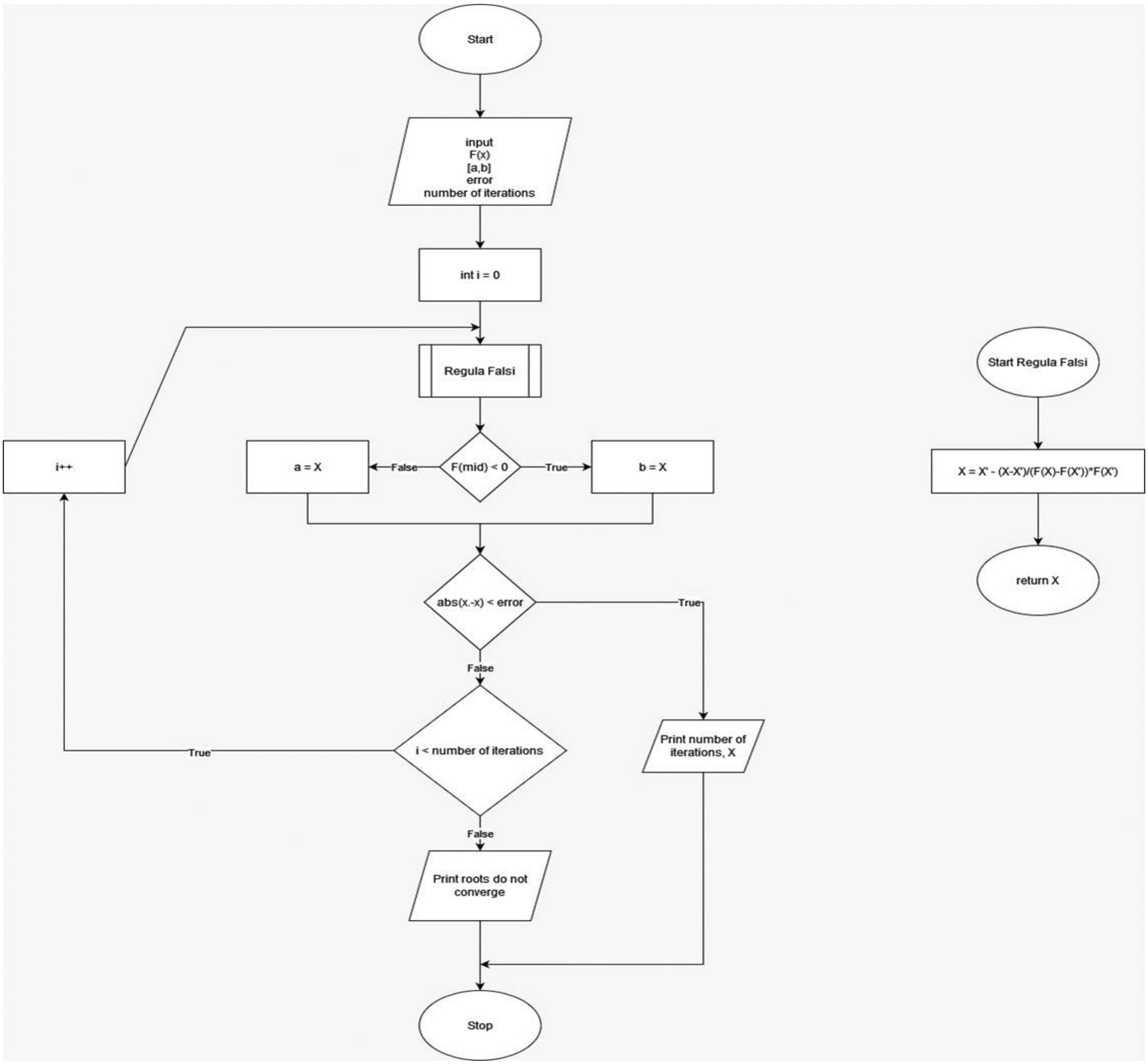
5. Do

```
        x2 = x0 - ((x0-x1) * f(x0))/(f(x0) - f(x1))

        If f(x0)*f(x2) < 0

                x1 = x2

         Else x0 = x2

         End If

 While abs(f(x2) > e
```

6. Print root as x2

7. Stop

Flow Charts:



A) f(x) =X.^2-4

Tolerance =0.001

Interval = [0,3]

Step	X-Lower	X-Upper	X-Root	F(Xr)	Error %
1	0	3	1.333333333	-2.222222222	-
2	1.333333333	3	1.846153846	-0.5917159763	27.77777778
3	1.846153846	3	1.968253968	-0.1259763165	6.203473945
4	1.968253968	3	1.993610224	-0.02551827619	1.271876272
5	1.993610224	3	1.998720409	-0.005116724772	0.2556728696
6	1.998720409	3	1.999744016	-0.001023868941	0.05118689720
7	1.999744016	3	1.999948801	-2.047947572e-4	0.01023947575
8	1.999948801	3	1.999989760	-4.095979029e-5	0.002047979029

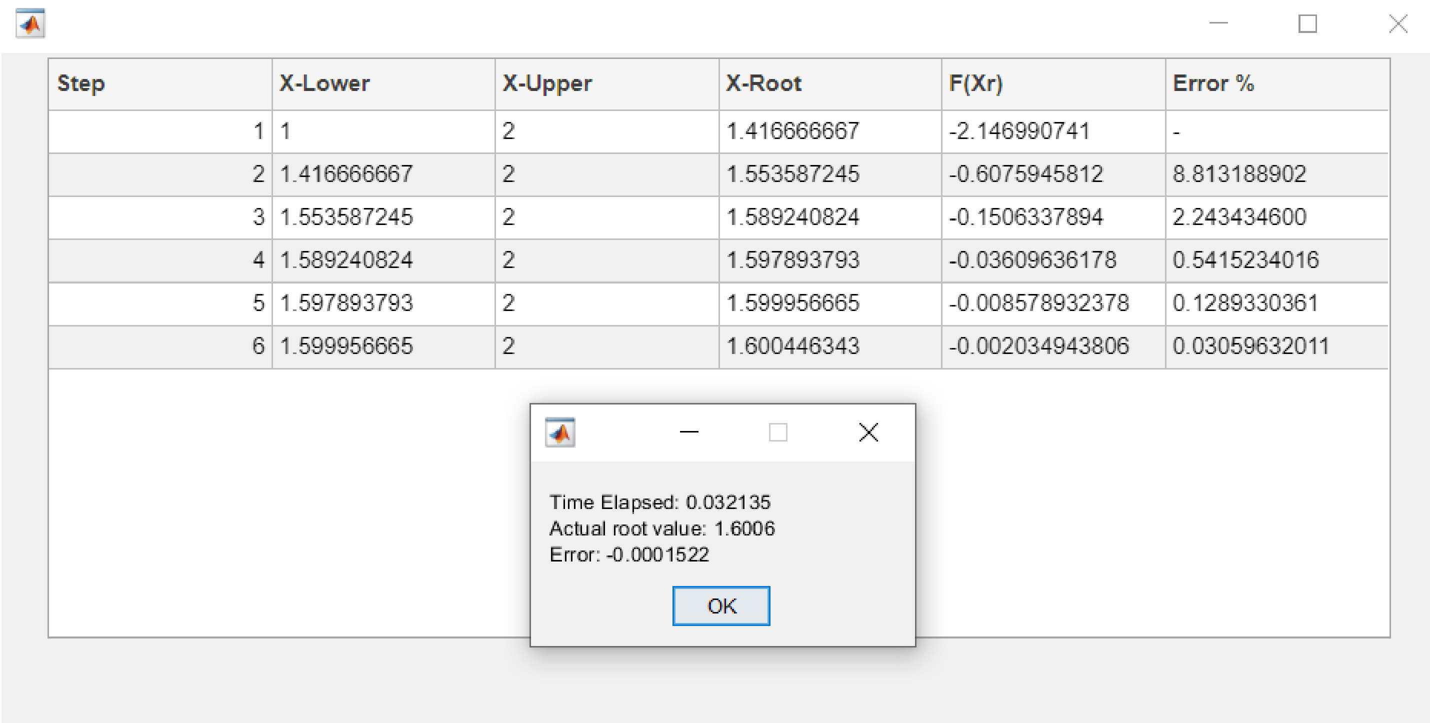
Time Elapsed: 0.025766
Actual root value: 2
Error: -1.024e-05

OK

B) f(x)=2x3-2x-5

Tolerance =0.001

Interval = [1,2]



Step	X-Lower	X-Upper	X-Root	F(Xr)	Error %
1	1	2	1.416666667	-2.146990741	-
2	1.416666667	2	1.553587245	-0.6075945812	8.813188902
3	1.553587245	2	1.589240824	-0.1506337894	2.243434600
4	1.589240824	2	1.597893793	-0.03609636178	0.5415234016
5	1.597893793	2	1.599956665	-0.008578932378	0.1289330361
6	1.599956665	2	1.600446343	-0.002034943806	0.03059632011

c) f(x) = e^x + 2x + 2 cos (x) -6

Tolerance = 0.001

Interval= [1.8,1.9]

Step	X-Lower	X-Upper	X-Root	F(Xr)	Error %
1	2.5	2.6	2.543288644	-0.01772286218	-
2	2.543288644	2.6	2.545059151	-7.123835766e-4	0.06956644701
3	2.545059151	2.6	2.545130226	-2.857752421e-5	0.002792579465
4	2.545130226	2.6	2.545133077	-1.146305594e-6	1.120193865e-4

Time Elapsed: 0.10399
Actual root value: 2.5451
Error: -1.1914e-07

OK

```
False Position method has converged
```

3)Fixed point :

<u>The algorithm we used:</u>

1. Start

2. Define function as f(x)

3. Define convergent form g(x)

4. Input: a. Initial guess x0 b. Tolerable Error e c. Maximum Iteration N

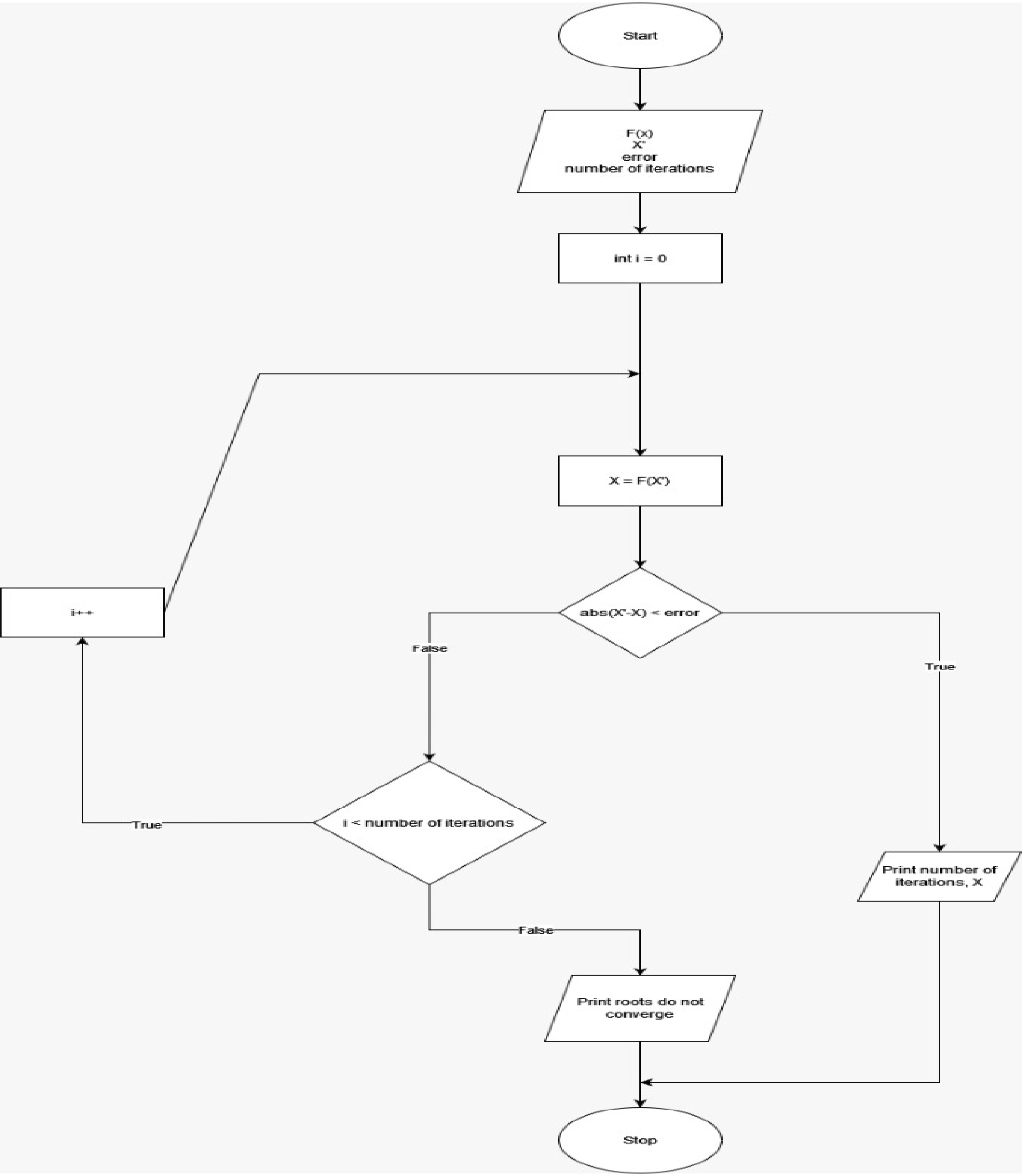
5. Initialize iteration counter: step = 1

6. Do

```
        x1 = g(x0) step = step + 1
          If step > N
              Print "Not Convergent" Stop
          End If
         x0 = x1
 While abs f(x1) > e
```

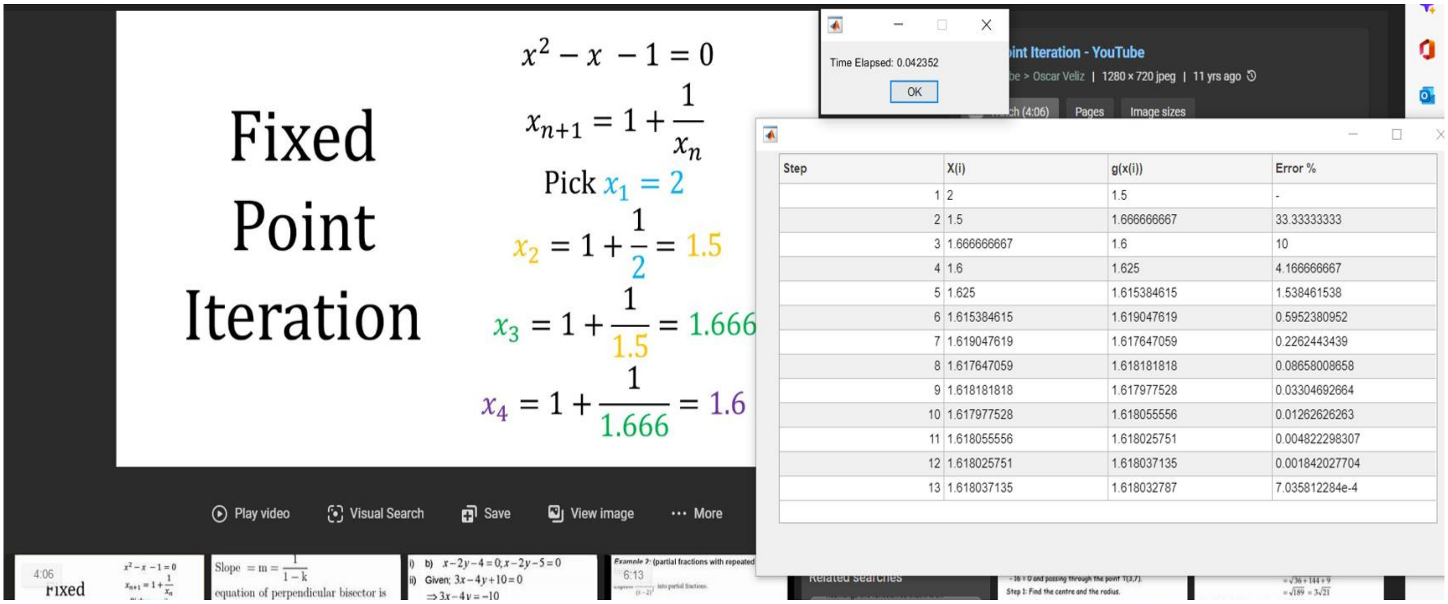
7. Print root as x1

8. Stop


Start
F(x)
X'
error
number of iterations
int i = 0
X = F(X')
abs(X'-X) < error
False
True
i++
i < number of iterations
True
False
Print number of iterations, X
Print roots do not converge
Stop

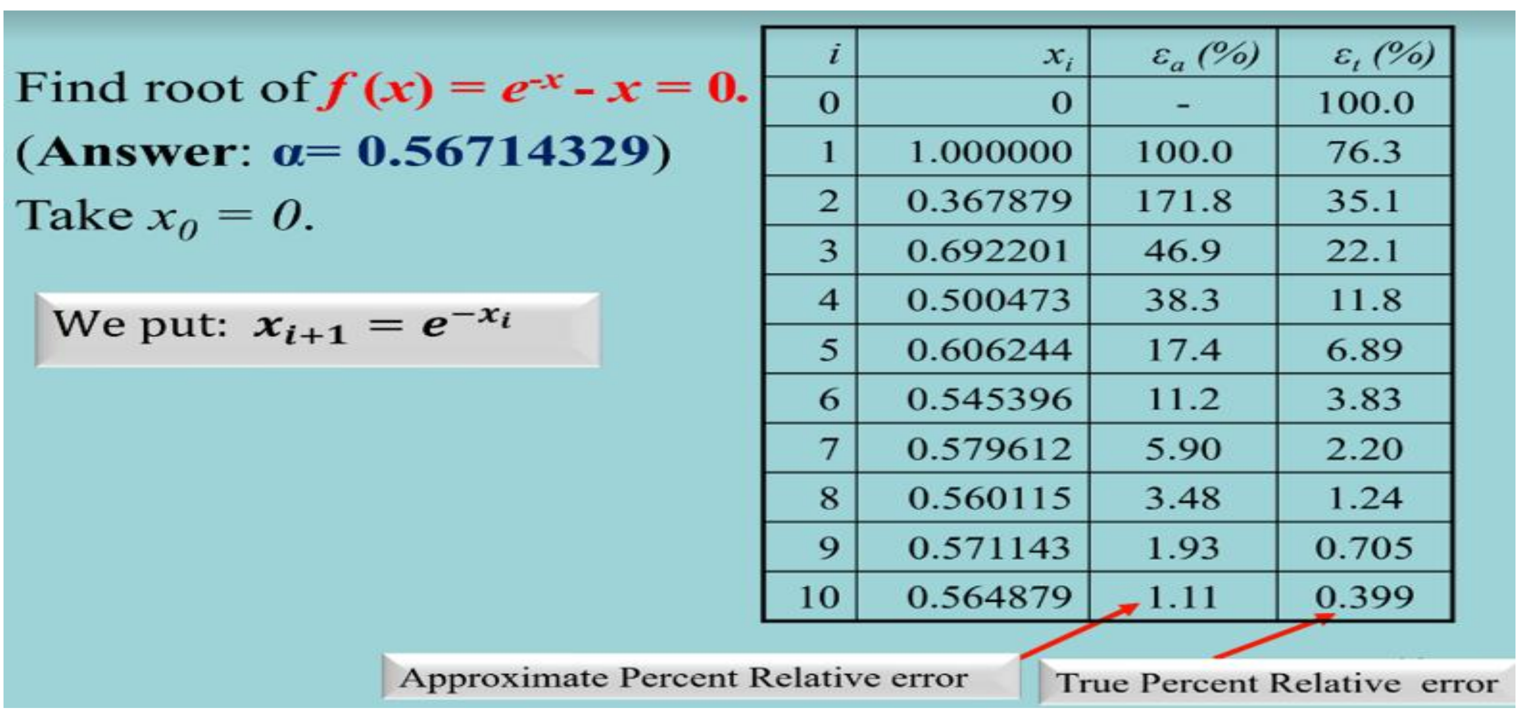
a) F(x)=x^2-x-1

Interval[1,3]



Step	X(i)	g(x(i))	Error %
1	2	1.5	-
2	1.5	1.666666667	33.33333333
3	1.666666667	1.6	10
4	1.6	1.625	4.166666667
5	1.625	1.615384615	1.538461538
6	1.615384615	1.619047619	0.5952380952
7	1.619047619	1.617647059	0.2262443439
8	1.617647059	1.618181818	0.08658008658
9	1.618181818	1.617977528	0.03304692664
10	1.617977528	1.618055556	0.01262626263
11	1.618055556	1.618025751	0.004822298307
12	1.618025751	1.618037135	0.001842027704
13	1.618037135	1.618032787	7.035812284e-4

b) f(x)=e^-x – x   interval =[0,1]



i	x_i	ε_a (%)	ε_t (%)
0	0	-	100.0
1	1.000000	100.0	76.3
2	0.367879	171.8	35.1
3	0.692201	46.9	22.1
4	0.500473	38.3	11.8
5	0.606244	17.4	6.89
6	0.545396	11.2	3.83
7	0.579612	5.90	2.20
8	0.560115	3.48	1.24
9	0.571143	1.93	0.705
10	0.564879	1.11	0.399

Time Elapsed: 0.020512

OK

Step	X(i)	g(x(i))	Error %
1	0	1	-
2	1	0.3678794412	100
3	0.3678794412	0.6922006276	171.8281828
4	0.6922006276	0.5004735006	46.85363946
5	0.5004735006	0.6062435351	38.30914659
6	0.6062435351	0.5453957860	17.44678968
7	0.5453957860	0.5796123355	11.15662253
8	0.5796123355	0.5601154614	5.903350814
9	0.5601154614	0.5711431151	3.480866980
10	0.5711431151	0.5648793474	1.930803931
11	0.5648793474	0.5684287250	1.108868242
12	0.5684287250	0.5664147331	0.6244191192
13	0.5664147331	0.5675566373	0.3555684138
14	0.5675566373	0.5669089119	0.2011965161
15	0.5669089119	0.5672762322	0.1142556402
16	0.5672762322	0.5670678984	0.06475156780
17	0.5670678984	0.5671860501	0.03673877244
18	0.5671860501	0.5671190401	0.02083120848
19	0.5671190401	0.5671570440	0.01181586888
20	0.5671570440	0.5671354902	0.006700779702
21	0.5671354902	0.5671477143	0.003800466624
22	0.5671477143	0.5671407815	0.002155356274
23	0.5671407815	0.5671447133	0.001222412855
24	0.5671447133	0.5671424834	6.932777789e-4

4) **Newton Raphson**:

The algorithm we used:

1. Start

2. Define function as f(x)

3. Define derivative of function as g(x)

4. Input: a. Initial guess x0 b. Tolerable Error e c. Maximum Iteration N

5. Initialize iteration counter step = 1

6. Do

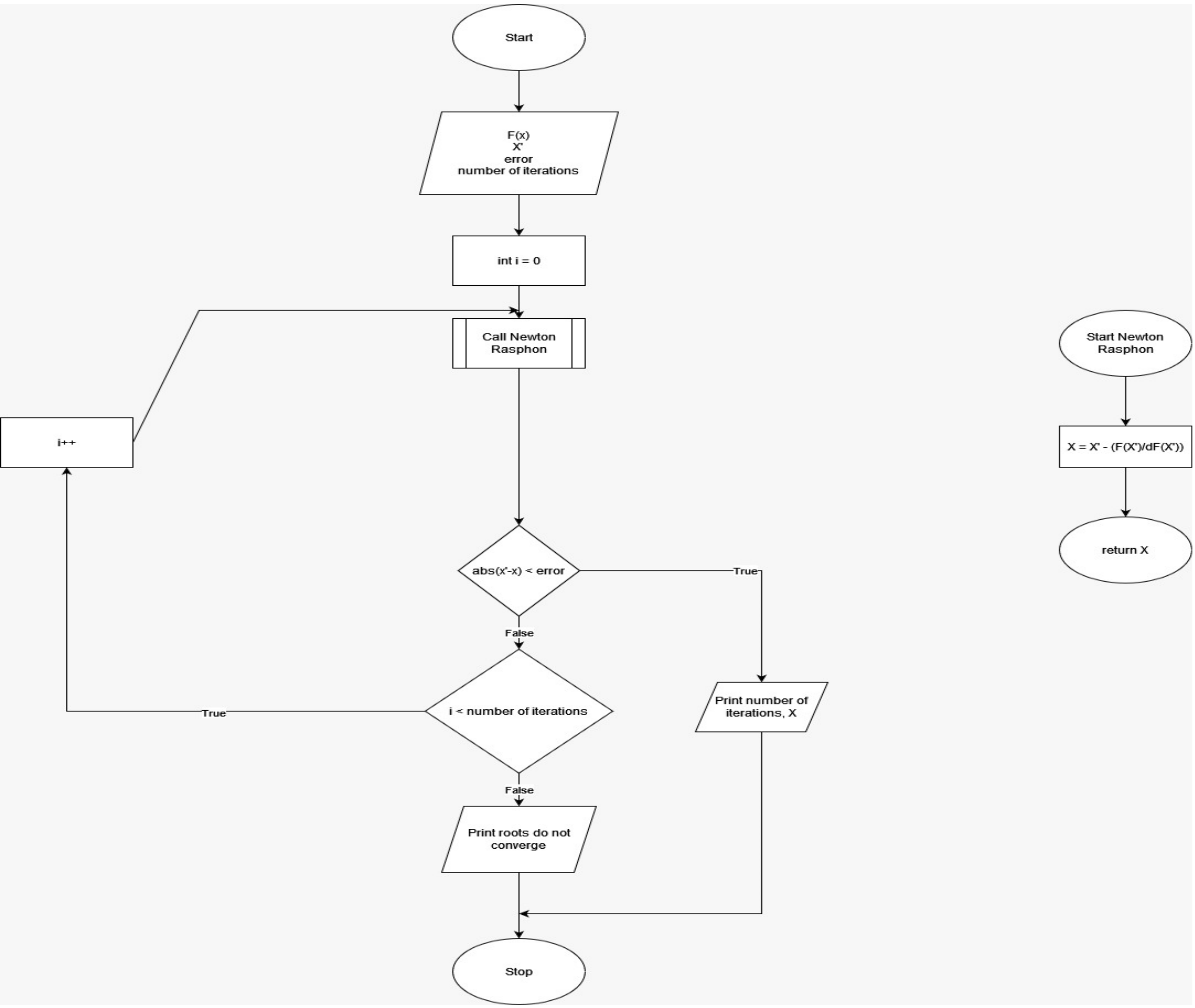
```
    If g(x0) = 0
        Print "Mathematical Error" Stop
    End If
      x1 = x0 - f(x0) / g(x0)
      x0 = x1 step = step + 1
      If step > N
      Print "Not Convergent" Stop
```

End If

While abs f(x1) > e

7. Print root as x1

8. Stop



A) f(x)=x − 0. 75 − 0. 2sin (x)

Tolerance = 10^−1

Interval = [0, π/2]

Xo= π/4

Step	X(i)	F(Xi)	F'(Xi)	X(i+1)	Error%
1	0.785398	-0.1060233331	0.8585786207	0.9088850408	13.58665125
2	0.9088850408	0.001121251243	0.8770748735	0.9076066422	0.1408538112
3	0.9076066422	1.288740830e-7	0.8768732890	0.9076064952	1.619314302e-5

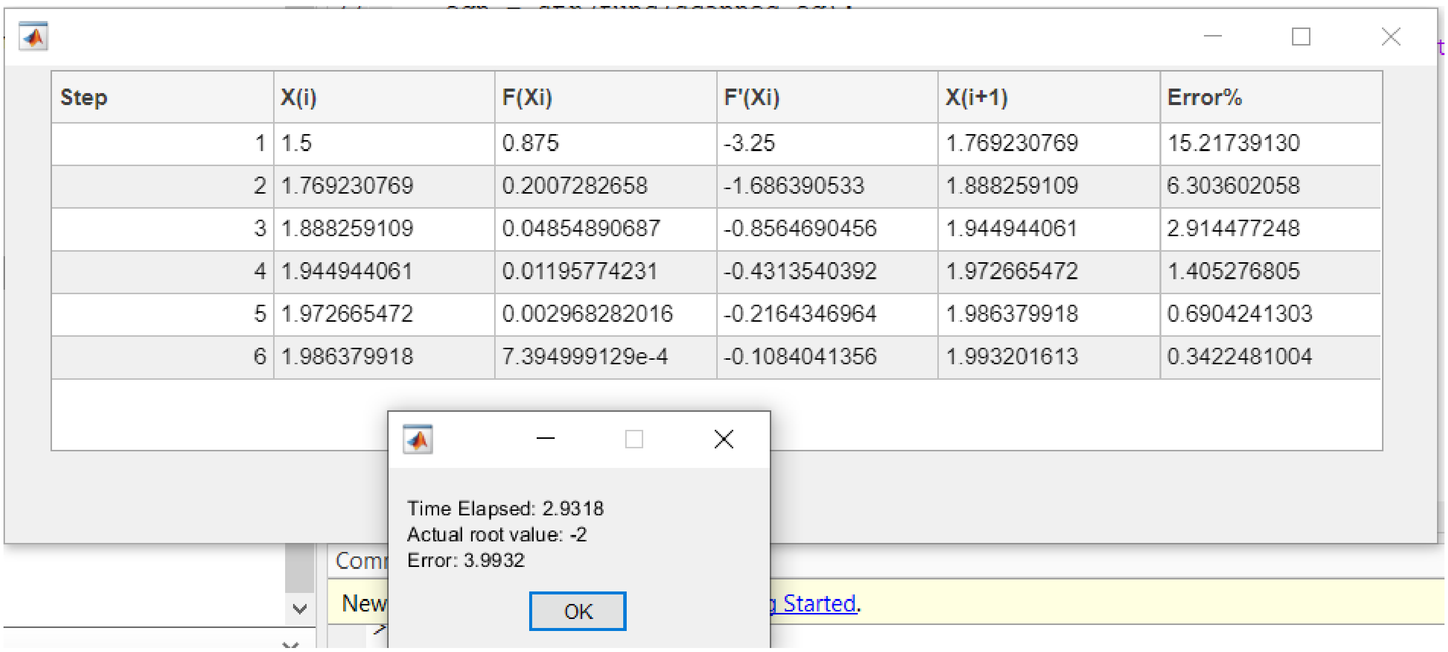
Time Elapsed: 10.3084
Actual root value: 0.90761
Error: 1.9984e-15

OK

B)f(x)=x3− 2x2− 4x + 8

Tolerance = 0.01

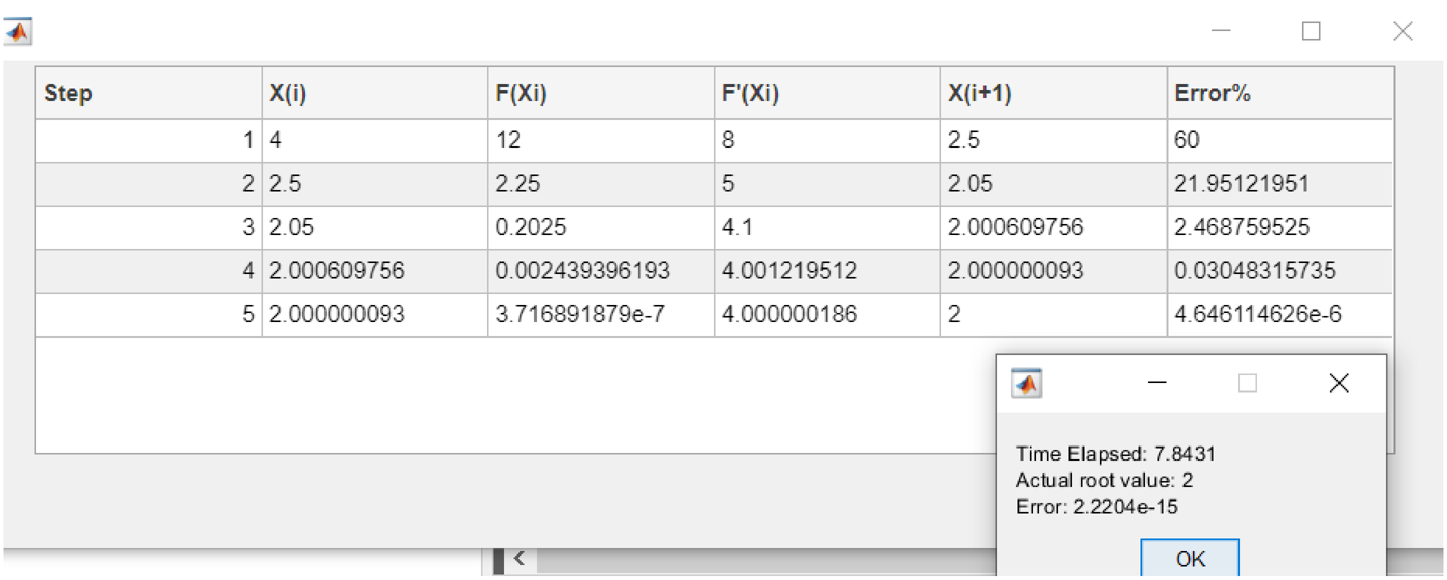
Xo = 1.5



Step	X(i)	F(Xi)	F'(Xi)	X(i+1)	Error%
1	1.5	0.875	-3.25	1.769230769	15.21739130
2	1.769230769	0.2007282658	-1.686390533	1.888259109	6.303602058
3	1.888259109	0.04854890687	-0.8564690456	1.944944061	2.914477248
4	1.944944061	0.01195774231	-0.4313540392	1.972665472	1.405276805
5	1.972665472	0.002968282016	-0.2164346964	1.986379918	0.6904241303
6	1.986379918	7.394999129e-4	-0.1084041356	1.993201613	0.3422481004

c) f(x)=x^2-4

Xo = 4



Step	X(i)	F(Xi)	F'(Xi)	X(i+1)	Error%
1	4	12	8	2.5	60
2	2.5	2.25	5	2.05	21.95121951
3	2.05	0.2025	4.1	2.000609756	2.468759525
4	2.000609756	0.002439396193	4.001219512	2.000000093	0.03048315735
5	2.000000093	3.716891879e-7	4.000000186	2	4.646114626e-6

5) Secant:

The algorithm we used:

1. Start

2. Define function as f(x)

3. Input: a. Initial guess x0, x1 b. Tolerable Error e c. Maximum Iteration N

4. Initialize iteration counter step = 1

5. Do If f(x0) = f(x1)

Print "Mathematical Error"

Stop

End If

x2 = x1 - (x1 - x0) * f(x1) / (f(x1) - f(x0))

x0 = x1

x1 = x2

step = step + 1 If step > N Print "Not Convergent"

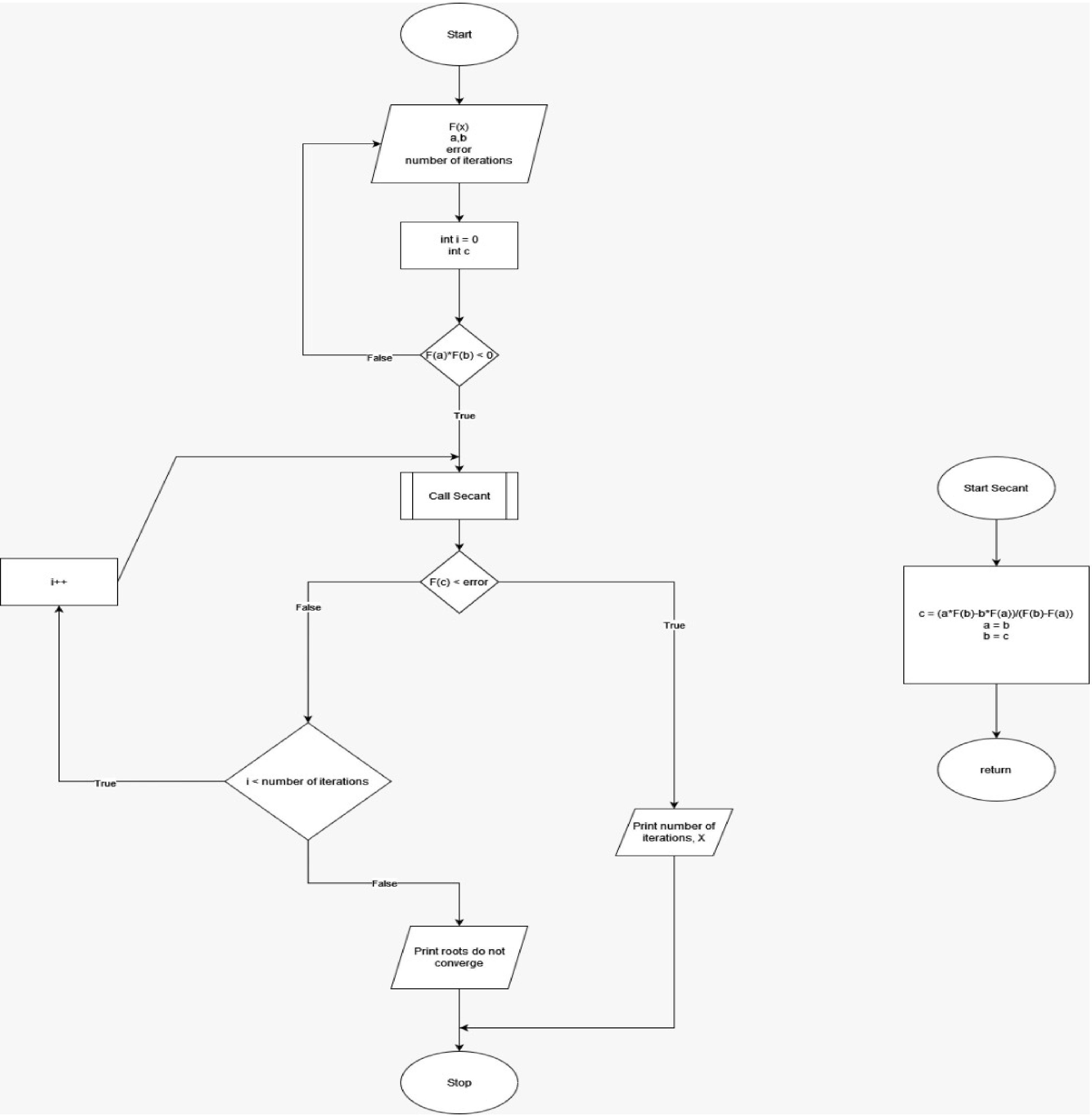
Stop

End If

While abs f(x2) > e

6. Print root as x2

7. Stop



Start
F(x)
a,b
error
number of iterations
int i = 0
int c
F(a)*F(b) < 0
False
True
Call Secant
F(c) < error
False
True
i++
i < number of iterations
True
False
Print number of iterations, X
Print roots do not converge
Stop
Start Secant
c = (a*F(b)-b*F(a))/(F(b)-F(a))
a = b
b = c
return

A)

f(x) = 3x^4+ 6. 1x^3− 2x^2+3x + 2

Interval = [-1, -3]

Tolerance =10 ^-2

Step	X(i-1)	X(i)	F(X(i-1))	F(X(i))	X(i+1)	Error%
1	-1	-3	-6.1	53.3	-1.205387205	-148.8826816
2	-3	-1.205387205	53.3	-8.872216564	-1.461485415	-17.52314509
3	-1.205387205	-1.461485415	-8.872216564	-12.01165338	-0.4816399503	-203.4394083
4	-1.461485415	-0.4816399503	-12.01165338	-0.4289832965	-0.4453497568	-8.148695009
5	-0.4816399503	-0.4453497568	-0.4289832965	-0.1535174205	-0.4251251941	-4.757319253
6	-0.4453497568	-0.4251251941	-0.1535174205	-0.007531149433	-0.4240818480	-0.2460246971
7	-0.4251251941	-0.4240818480	-0.007531149433	-1.452235312e-4	-0.4240613336	-0.004837620113

Time Elapsed: 0.0071127
Actual root value: -2.4305
Error: 2.0064

OK

B)f(x) =x^5 - 5 x + 3

Interval = [0.5,1.5]

Tolerance = 10^-3

Step	X(i-1)	X(i)	F(X(i-1))	F(X(i))	X(i+1)	Error%
1	0.5	1.5	0.53125	3.09375	0.2926829268	412.5
2	1.5	0.2926829268	3.09375	1.538733132	-0.9019914327	-132.4485263
3	0.2926829268	-0.9019914327	1.538733132	6.912905294	0.6347421246	242.1036036
4	-0.9019914327	0.6347421246	6.912905294	-0.07067518890	0.6191900830	2.511674858
5	0.6347421246	0.6191900830	-0.07067518890	-0.004933950793	0.6180228855	0.1888599141
6	0.6191900830	0.6180228855	-0.004933950793	4.741677498e-5	0.6180339959	0.001797692490

Time Elapsed: 0.0001354
Actual root value: 1.2757
Error: -0.65765

OK

```
>> Main
Converged
>> Main
```

C) f(x)=x3-x-1

Interval = [1,2]

Tolerance=10^-4

Step	X(i-1)	X(i)	F(X(i-1))	F(X(i))	X(i+1)	Error%
1	1	2	-1	5	1.166666667	71.42857143
2	2	1.166666667	5	-0.5787037037	1.253112033	6.898454746
3	1.166666667	1.253112033	-0.5787037037	-0.2853630296	1.337206446	6.288812988
4	1.253112033	1.337206446	-0.2853630296	0.05388058668	1.323850096	1.008901951
5	1.337206446	1.323850096	0.05388058668	-0.003698115440	1.324707937	0.06475692647
6	1.323850096	1.324707937	-0.003698115440	-4.273426281e-5	1.324717965	7.570533496e-4
7	1.324707937	1.324717965	-4.273426281e-5	3.458221465e-8	1.324717957	6.121414230e-7

Time Elapsed: 0.0005262
Actual root value: 1.3247
Error: -7.5717e-14

```
>> Main
```

From previous examples verifies This: