Greg Wagner

Lab 7

Numerical Analysis

Apr 4, 2019

The objective of this lab was to create a C++ program that implemented two methods for finding roots of a function: Secant and Bisection. Given the function $f(x) = x^3 - 2x^2 - 5x + 6$. The methods were then used to evaluate possible roots over the intervals given below. This table contains the iterative results of the programs output. For a more detailed form of this information please run main.cpp in project Wagner7. We can see that the secant method is superior to the bisection method because it can correct for poor initial guesses.

interval [0.00 - 4.00]

Secant Method

Iteration	Approx. root	x_tolerance	y_tolerance
1	0	0	2
2	-2	-2	0

Bisection Method

Not able to find roots on interval: bad guesses...

interval [0.00 - 2.00]

Secant Method

Iteration	Approx. root	x_tolerance	y_tolerance
1	2.000000	2.000000	0.800000
2	1.200000	1.200000	0.323596
3	0.876404	0.876404	-0.128111
4	1.004515	1.004515	0.004435
5	1.000081	1.000081	0.000081
6	1.000000	1.000000	0.000000

Bisection Method

Iteration	Approx. root	x_tolerance	y_tolerance
1	1	1	0

interval [2.00 - 4.00]

Secant Method

Iteration	Approx. root	x_tolerance	y_tolerance
1	2.000000	2.000000	-0.363636
2	2.363636	2.363636	-6.453265
3	2.363636	2.363636	-0.049303
4	2.412939	2.412939	-1.430303
5	2.412939	2.412939	-0.296298
6	2.709237	2.709237	-0.525330
7	2.709237	2.709237	-0.241828
8	2.951065	2.951065	-0.061205
9	3.012270	3.012270	0.012699
10	2.999571	2.999571	-0.000425
11	2.999996	2.999996	-0.000004

Bisection Method

Iteration	Approx. root	x_tolerance	y_tolerance
1	3	1	0

interval [0.00 - 3.00]

Secant Method
exact root found at
3.000000

Bisection Method
exact root found at
3.000000

interval [1.00 - 2.00]

Secant Method
exact root found at
1.000000

Bisection Method
exact root found at
1.000000

interval [2.00 - 30.00]

Secant Method

Iteration	Approx. root	x_tolerance	y_tolerance

1	2.000000	2.000000	-0.004469
2	2.000000	2.000000	4.072893
3	-2.072893	-2.072893	1.616075
4	-2.072893	-2.072893	-0.035425
5	-2.037468	-2.037468	-0.036074
6	-2.001394	-2.001394	-0.001366
7	-2.000027	-2.000027	-0.000027
8	-2.000000	-2.000000	0.000000

Bisection Method

Iteration	Approx. root	x_tolerance	y_tolerance
1	16.000000	14.000000	3510.000000
2	9.000000	7.000000	528.000000
3	5.500000	3.500000	84.375000
4	3.750000	1.750000	11.859375
5	2.875000	0.875000	-1.142578
6	3.312500	0.437500	3.839111
7	3.093750	0.218750	0.999847
8	2.984375	0.109375	-0.154545
9	3.039063	0.054688	0.401366
10	3.011719	0.027344	0.118150
11	2.998047	0.013672	-0.019505
12	3.004883	0.006836	0.048995
13	3.001465	0.003418	0.014663
14	2.999756	0.001709	-0.002441
15	3.000610	0.000854	0.006106
16	3.000183	0.000427	0.001831
17	2.999969	0.000214	-0.000305
18	3.000076	0.000107	0.000763
19	3.000023	0.000053	0.000229
20	2.999996	0.000027	-0.000038

interval	[10.00 - 30.00]

Secant Method

Iteration	Approx. root	x_tolerance	y_tolerance
1	10.000000	10.000000	0.622222
2	9.377778	9.377778	2.554716
3	6.823061	6.823061	1.219381
4	5.603681	5.603681	1.055660
5	4.548021	4.548021	0.688077
6	3.859943	3.859943	0.460108

7	3.399836	3.399836	0.258272
8	3.141564	3.141564	0.111131
9	3.030433	3.030433	0.027681
10	3.002752	3.002752	0.002694
11	3.000058	3.000058	0.000057
12	3.000000	3.000000	0.000000

Bisection Method

Not able to find roots on interval: bad guesses...

Learned items…

- Bisection method
- Newtons method
- Secant Method
- C++ practice