Yitian Huang
Mar 11, 2022
CS 3010
Professor RodrÍguez

Assignment 3, finding root

Exercise 1:

1) f(x) = x^3 + 3x - 1, on [0,1]

f(0) = 0 + 0 - 1 = -1 f(1) = 1 + 3 - 1 = 3 f(1/2) = 1/8+3/2-1 = 5/8 f(1/4) = 1/64 + 3/4 -1 = -15/64 f(3/8) = 27/512 + 9/8 - 1 = 91/512 f(5/16) = 125/4096 + 15/16 - 1 = -131/4096 f(11/32) = 1331/32768 + 33/32 - 1 = 2355/32768 f(21/64) = 9261/262144 + 63/64 - 1 = 5165/262144 f(41/128) = 68921/2097152 + 123/128 - 1 = -12999/2097152 f(83/256) = 571787/1677216 + 249/256 - 1 = 113038/1677216 f(165/512) = 4492125/134217728 + 495/512 - 1 = 35677/134217728 f(329/1024) = 35611289/1073741824 = -3186023/1073741824	iteration 1: f(0) * f(1) < 0, (0+1)/2 = 1/2 iteration 2: f(0) * f(1/2) < 0, (0+1/2)/2 = 1/4 iteration 3: f(0) * f(1/4) > 0, (1/41/2)/2 = 3/8 iteration 4: f(1/4) * f(3/8) < 0, (1/4+3/8)/2 = 5/16 iteration 5: f(1/4) * f(5/16) > 0, (5/16+3/8)/2=11/32 iteration 6: f(5/16)* f(11/32) < 0, (5/16+11/32)/2=21/64 iteration 7: f(5/16)*f(21/64) < 0, (5/16+21/64)/2=41/128 iteration 8: f(5/16)*f(41/128) > 0, (21/64+41/128)/2=83/256 iteration 9: f(41/128)*f(83/256) < 0, (41/128+83/256)/2 = 165/512 iteration 10: f(41/128) * f(165/512) < 0, (41/128+165/512)/2 = 329/1024

After 10 iterations, the root is approximate at x = 329/1024, f(x) = -3186023/1073741824 ≈ -0.002967, but during one of the iteration, the closest root it found is actually x = 165/512, f(x) 35677/134217728 ≈ -0.000266

2) g(x) = x^3 - 2 sin x, on [0.5,2]

f(0.5) = 0.5^3 - 2sin(0.5) ≈ -0.8338 f(2) = 2^3 - 2sin(2) ≈ 6.1814 f(1.25) = 1.25^3 - 2sin(1.25) ≈ 0.0552 f(0.875) = 0.875^3 - 2sin(0.875) ≈ -0.8652 f(1.0625) = 1.0625^3 - 2sin(1.0625) ≈ -0.5477 f(1.15625) = 1.15625^3 - 2sin(1.15625) ≈ -0.2848 f(1.203125) = 1.203125^3 - 2sin(1.203125) ≈ -0.1280 f(1.2265625) = 1.2265625^3 - 2sin(1.2265625) ≈ -0.03736 f(1.23828125) = 1.23828125^3 - 2sin(1.23828125) ≈ 0.008258 f(1.232421875) = 1.232421875^3 - 2sin(1.232421875) ≈ -0.01471 f(1.2353515625) = 1.2353515625^3 - 2sin(1.2353515625) ≈ -0.003266 f(1.2368164375) = 1.2368164375^3 -	iteration 1: f(0.5)*f(2)<0, (0.5+2)/2=1.25 iteration 2: f(0.5)*f(1.25)<0, (0.5+1.25)/2=0.875 iteration 3: f(0.5)*f(0.875)>0, (0.875+1.25)/2=1.0625 iteration 4: f(0.875)*f(1.0625)>0, (1.0625+1.25)/2=1.15625 iteration 5: f(1.0625)*f(1.15625)>0, (1.15625+1.25)/2=1.203125 iteration 6: f(1.15625)*f(1.203125)>0, (1.203125+1.25)/2=1.2265625 iteration 7: f(1.203125)*f(1.2265625)>0, (1.2265625+1.25)/2=1.23828125 iteration 8: f(1.2265625)*f(1.23828125)<0, (1.2265625+1.23828125)/2=1.232421875 iteration 9: f(2265625)*f(1.232421875)>0, (1.232421875+1.23828125)/2=1.2353515625 iteration 10: f(1.232421875)*f(1.2353515625)>0, (1.2353515625+1.23828125)/2= 1.2368164375

2sin(1.2368164375) ≈ 0.0024861348	

After 10 iterations, the root is approximate at x = 1.2368164375, f(x) ≈ 0.0024861348.

3) h(x) = x + 10 - x cosh(50/x), on [120,130]

f(120) = 120+10-120cosh(50/120) ≈ -0.56825 f(130) = 130+10-130cosh(50/130) ≈ 0.26550 f(125) = 125+10-125cosh(50/125) ≈ -0.13405 f(127.5) = 127.5+10-127.5cosh(50/127.5) ≈ 0.06790 f(126.25) = 126.25+10-126.25cosh(50/126.25) ≈ -0.0310806 f(126.875) = 126.875 + 10 - 126.875cosh(50/126.875) ≈ 0.019612 f(126.5625) = 126.5625+10-126.5625cosh(50/126.5625) ≈ -0.005669 f(126.71875) = 126.71875+10-126.71875cosh(50/126.71875) ≈ 0.0069878 f(126.640625) = 126.640625+10-126.640625cosh(50/126.640625) ≈ 0.0006633 f(126.6015625) = 126.6015625+10-126.6015625cosh(50/126.6015625) ≈ -0.0025018 f(126.62109375) = 126.62109375 +10 - 126.62109375 cosh(50/126.62109375) ≈ 0.0009189843 f(126.630859375) = 126.630859375+10-126.630859375cosh(50/126.630859375) ≈ -1.277356x10^-4	iteration 1: f(120)*f(130)<0, (120+130)/2= 125 iteration 2: f(120)*f(125)>0, (125+130)/2= 127.5 iteration 3: f(125)*f(127.5)<0, (125+127.5)/2= 126.25 iteration 4: f(125)*f(126.25)>0, (126.25+127.5)/2= 126.875 iteration 5: f(126.25)*f(126.875)<0, (126.25+126.875)/2= 126.5625 iteration 6: f(126.25)*f(126.5625)>0, (126.5625+1126.875)/2= 126.71875 iteration 7: f(126.5625)*f(126.71875)<0, (126.5625+126.71875)/2= 126.640625 iteration 8: f(126.5625)*f(126.640625)<0, (126.5625+126.640625)/2= 126.6015625 iteration 9: f(126.5625)*f(126.6015625)>0, (126.6015625+126.640625)/2=126.62109375 iteration 10: f(126.6015625)*f(126.62109375)>0, (126.62109375+126.640625)/2= 126.630859375

After 10 iterations, the root is approximate at x = 126.630859375, f(x) ≈ -1.277356x10^-4.

Exercise 2:

f(x) = x^3 + 2x^2 + 10x - 20, starting with x_0 = 2

f’(x) = 3x^2 + 4x + 10

x_i+1 = x_i - f(x_i)/f’(x_i)

x_0 = 2 x_1 = 1.4666667 x_2 = 1.371512 x_3 = 1.3688102 x_4 = 1.3688081 x_5 = 1.368808107 x_6 =1.3688081078 x_7 =1.36880810782 x_8 =1.368808107821 x_9 =1.3688081078213	iteration 1: f(2) = 16 != 0, x_1 = x_0 - f(x_0)/f(x_0) = 2 - 16/30 ≈ 1.4666667 iteration 2: f(1.4666667) = 2.1238517 != 0 , x_2 = 1.4666667 - 2.21238517/22.32 ≈ 1.371512 iteration 3: f(1.371512) = 0.05708635 != 0, x_3 = 1.371512 - 0.05708635/21.129183 = 1.3688102 iteration 4: f(1.3688102) = 4.143692x10^-5 != 0, x_4 = 1.3688102 - 4.143692x10^-5/21.096165 = 1.3688081 iteration 5: f(1.3688081) = -1.650008x10^-7 != 0, x_5 = 1.3688081 - -1.650008x10^-7/21.096139 = 1.368808107 iteration 6: x_6 = 1.368808107 - (-1.732779x10^-8)/21.096139)=

x_10=1.36880810782137	1.3688081078 iteration 7: x_7=1.3688081078-(-4.50882x10^-10/21.096139) = 1.36880810782 iteration 8: x_8=1.36880810782-(-2.895462x10^-11/21.096139) = 1.368808107821 iteration 9: x_9=1.368808107821-(-7.862155x10^-12/21.096139) = 1.3688081078213 iteration 10: x_10= 1.3688081078213 - (-1.534772x10^-12/21.096139) = 1.36880810782137

After 10 iterations x = 1.36880810782137, f(x) = -5.329071 x 10^-14

Exercise 3:
f(x) = x^3 + 2x^2 + 10x -20, with x_0 = 2, and x_1 = 1.
x_i+1 = x_i - ((x_i - x_i-1)f(x_i))/(f(x_i)-f(x_i-1))
f(x_0)=f(2)=2^3+2(2^2)+10(2)-20=16
f(x_1)=f(1)=1^3+2(1^2)+10(1)-20=-7

x_0 = 2 x_1 = 1 x_2 = 1.30435 x_3 = 1.3760534 x_4 = 1.36867196 x_5 = 1.36880782 x_6 = 1.368808107 x_7 = 1.3688081078 x_8 = 1.36880810782 x_9 =1.368808107821 x_10 = 1.3688081078213 x_11 = 1.36880810782137	iteration 1: x_2 = x_1 - ((x_1-x_0)f(x_1))/(f(x_1)-f(x_0)) = 1-((1-2)*-7)/(-7-16) = 1.30435 iteration 2: f(1.30435) = -1.3347138 != 0. x_3 = 1.30435-((1.30435 - 1)*-1.3347138)/(-1.3347138- -7) = 1.3760534 iteration 3: f(1.3760534) = 0.15316863 != 0. x_4 = 1.3760534-((1.3760534-1.30435)*0.15316863)/(0.15316863- -1.3347138) = 1.36867196 iteration 4: f(1.36867196) = -0.0028720802 != 0. x_5 = 1.36867196 -((1.36867196-1.3760534)*-0.0028720802)/(-0.0028720802-0.15316863) = 1.36880782 iteration 5: f(1.36880782) = -6.071919x10^-6 != 0. x_6 = 1.36880782-((1.36880782-1.36867196)*-6.071919x10^-6)/(-6.071919x10^-6--0.0028720802) = 1.368808107 iteration 6: f(1.368808107) = -1.732779 x 10^-8 != 0. x_7 = 1.368808107 - ((1.368808107-1.36880782)*-1.732779 x 10^-8)/(-1.732779 x 10^-8-(-6.071919x10^-6)) = 1.3688081078 iteration 7: f(1.3688081078) =-4.50882x10^-10 x_8 = 1.3688081078 - ((1.3688081078 - 1.368808107)*-4.50882x10^-10)/(-4.50882x10^-10-(-1.732779 x 10^-8)) = 1.36880810782 iteration 8: f(1.36880810782) = -2.895462x10^-11 x_9 = 1.36880810782 - ((1.36880810782-1.3688081078)*-2.895462x10^-11)/(-2.895462x 10^-11 - (-4.50882x10^-10)) = 1.368808107821 iteration 9: f(1.368808107821) = -7.862155x10^-12 x_10 = 1.368808107821 - ((1.368808107821-1.36880810782)*-7.862155x10^-12)/(-7.862155x10^-12-(-2.895462x10^-11)) = 1.3688081078213 iteration 10: f(1.3688081078213) = -1.534772x10^-12 x_11 = 1.3688081078213 -

	((1.3688081078213-1.368808107821)*-1.534772x10^-12)/(-1.534772x10^-12 - (-7.862155x10^-12)) = 1.36880810782137

After 10 iterations, x = 1.36880810782137 and f(x) = -5.329071 x 10^-14.

Exercise 4:
Input: fun1.pol
3
3 5 0 -7
Output:
py .\polyRoot.py 0 1 fun1.pol

```
PS C:\Users\ythua\vscode\polyRoot> py .\polyRoot.py 0 1 fun1.pol
Degree: 3 Polynomial: [3.0, 5.0, 0.0, -7.0]
Algorithm has converged after 24 iterations!
Root: 0.9451799988746643, Iteration: 24, Outcome: success
```

```
polyRoot.py    fun1.sol  ×
fun1.sol
1    0.9451799988746643 24 success
```

py .\polyRoot.py -newt 0 fun1.pol

```
PS C:\Users\ythua\vscode\polyRoot> py .\polyRoot.py -newt 0 fun1.pol
Degree: 3 Polynomial: [3.0, 5.0, 0.0, -7.0]
Small slope!
Root: 0, Iteration: 1, Outcome: fail
```

```
polyRoot.py    fun1.sol  ×
fun1.sol
1    0 1 fail
```

fail because the initial point is 0 and derivative, f'(0) is 0, which is bad, divide by 0 bad.
py .\polyRoot.py -sec -maxIter 100000 0 1 fun1.pol

```
PS C:\Users\ythua\vscode\polyRoot> py .\polyRoot.py -sec -maxIter 100000 0 1 fun1.pol
Degree: 3 Polynomial: [3.0, 5.0, 0.0, -7.0]
Algorithm has converged after 6 iterations!
Root: 0.9451800563938232, Iteration: 6, Outcome: success
```

```
polyRoot.py    fun1.sol  ×
fun1.sol
1    0.9451800563938232 6 success
```

Exercise 5:
f(x) = x^3 + 3x - 1, on [0,1]
Bisection

```
PS C:\Users\ythua\vscode\polyRoot> py .\polyRoot.py 0 1 fun2.pol
Degree: 3 Polynomial: [1.0, 0.0, 3.0, -1.0]
Algorithm has converged after 24 iterations!
Root: 0.32218533754348755, Iteration: 24, Outcome: success
```

Newton

```
PS C:\Users\ythua\vscode\polyRoot> py .\polyRoot.py -newt 1 fun2.pol
Degree: 3 Polynomial: [1.0, 0.0, 3.0, -1.0]
Algorithm has converged after 5 iterations!
Root: 0.3221853546260856, Iteration: 5, Outcome: success
```

Secant

```
PS C:\Users\ythua\vscode\polyRoot> py .\polyRoot.py -sec 0 1 fun2.pol
Degree: 3 Polynomial: [1.0, 0.0, 3.0, -1.0]
Algorithm has converged after 5 iterations!
Root: 0.3221852442945197, Iteration: 5, Outcome: success
```

Hybrid

```
PS C:\Users\ythua\vscode\polyRoot> py .\polyRoot.py -hybrid 0 1 fun2.pol
Degree: 3 Polynomial: [1.0, 0.0, 3.0, -1.0]
Max iteration reached without covergence with bisection method...
Switching to newton method
Algorithm has converged after 3 iterations!
Root: 0.3221853546260856, Iteration: 8, Outcome: success
```

f(x) = x^3 + 2x^2 + 10x - 20
bisection

```
PS C:\Users\ythua\vscode\polyRoot> py .\polyRoot.py 2 1 fun2.pol
Degree: 3 Polynomial: [1.0, 2.0, 10.0, -20.0]
Algorithm has converged after 24 iterations!
Root: 1.368808090686798, Iteration: 24, Outcome: success
```

newton

```
PS C:\Users\ythua\vscode\polyRoot> py .\polyRoot.py -newt 1 fun2.pol
Degree: 3 Polynomial: [1.0, 2.0, 10.0, -20.0]
Algorithm has converged after 4 iterations!
Root: 1.3688081078213745, Iteration: 4, Outcome: success
```

secant

```
PS C:\Users\ythua\vscode\polyRoot> py .\polyRoot.py -sec 2 1 fun2.pol
Degree: 3 Polynomial: [1.0, 2.0, 10.0, -20.0]
Algorithm has converged after 6 iterations!
Root: 1.3688081078326166, Iteration: 6, Outcome: success
```

hybrid

```
PS C:\Users\ythua\vscode\polyRoot> py .\polyRoot.py -hybrid 2 1 fun2.pol
Degree: 3 Polynomial: [1.0, 2.0, 10.0, -20.0]
Max iteration reached without covergence with bisection method...
Switching to newton method
Algorithm has converged after 3 iterations!
Root: 1.3688081078213727, Iteration: 8, Outcome: success
```