

bessel function J0와 nonlinear_func에 정의한 함수에 대해 아래와 같이 cpu time으로 convergence speed를 측정했다. bisection method, linear interpolation보다 open methods (newton-raphson, secant method)가 더 빠르게 동작함을 알 수 있다.

```
(base) lee@lee RootFinding % ./rootFinding
=====
Rough initial solutions for Bessel function J0 (bessj0) in [1.0, 10.0] are following brackets (zbrak):
[2.4040470123, 2.4049470425] / [5.5195488930, 5.5204486847] / [8.6529693604, 8.6538696289] /
=====

Exact solutions (rtbis):
2.4048259258 / 5.5200777054 / 8.6537284851 / Average cpu time used: 0.0000030000 sec
=====
Exact solutions (rtflsp):
2.4048256874 / 5.5200781822 / 8.6537275314 / Average cpu time used: 0.0000020000 sec
=====
Exact solutions (rtsec):
2.4048256874 / 5.5200781822 / 8.6537275314 / Average cpu time used: 0.0000016667 sec
=====
Exact solutions (rtnewt):
2.4048254490 / 5.5200781822 / 8.6537275314 / Average cpu time used: 0.0000016667 sec
=====
Exact solutions (rtsafe):
2.4048254490 / 5.5200781822 / 8.6537275314 / Average cpu time used: 0.0000020000 sec
=====
Exact solutions (muller):
2.4048254490 / 5.5200781822 / 8.6537275314 / Average cpu time used: 0.0000020000 sec
=====

Rough initial solutions for custom non-linear function (nonlinear_f) in [1.0, 5.0] are following brackets (zbrak):
[1.5707246065, 1.5711245537] / [3.1412572861, 3.1416573524] / [4.7123179436, 4.7127180099] /
=====

Exact solutions (rtsafe):
1.5707963705 / 3.1415927410 / 4.7123889923 / Average cpu time used: 0.0000016667 sec
=====
```