

[수치해석] Assignment2 report

이름: 권도현

학번: 2023065350

학과: 컴퓨터소프트웨어학부

Convergence speed 비교

먼저 **Root**가 있을 가능성이 높은 구간부터 출력해보았다.

```
Interval where root is likely to exist
Interval [a, b] a: 2.40400, b: 2.41300
Interval [a, b] a: 5.51798, b: 5.52698
Interval [a, b] a: 8.64992, b: 8.65892
```

이후, `muller` 를 포함하여 6가지 방법의 `Convergence speed` 를 `time` 라이브러리를 이용하여 측정해보았다.

```

-----
1. Find the roots using Bisection (rtbis.c)
   Root 1 is 2.40483
   Root 2 is 5.52008
   Root 3 is 8.65373
Average convergence speed is 0.0000030000

2. Find the roots using Linear Interpolation (rtflsp.c)
   Root 1 is 2.40483
   Root 2 is 5.52008
   Root 3 is 8.65373
Average convergence speed is 0.0000016667

3. Find the roots using Secant (rtsec.c)
   Root 1 is 2.40483
   Root 2 is 5.52008
   Root 3 is 8.65373
Average convergence speed is 0.0000010000

4. Find the roots using Newton-Raphson (rtnewt.c)
   Root 1 is 2.40483
   Root 2 is 5.52008
   Root 3 is 8.65373
Average convergence speed is 0.0000010000

5. Find the roots using Newton with bracketing (rtsafe.c)
   Root 1 is 2.40483
   Root 2 is 5.52008
   Root 3 is 8.65373
Average convergence speed is 0.0000010000

-----

Find the roots using Muller (muller.c)
   Root 1 is 2.40483
   Root 2 is 5.52008
   Root 3 is 8.65373
Average convergence speed is 0.0000013333
-----

```

- 시간을 소숫점 아래 10번째 자리까지 나타내었다.
- 실행할 때마다 Average convergence speed가 조금씩 달라졌다.
- 여러 번 실행하며 관찰한 결과, **Newton-Raphson = Newton with bracketing > Muller = Secant > Linear interpolation > Bisection** 순으로 빨랐다.
- 구간을 점점 줄여가는 **Bisection, Linear interpolation**보다 직전 추정점을 이용하는 **Open method (Muller, Secant, Newton)**이 수렴 속도가 빠른 경향이 있는 것을 확인할 수 있다.
- Open method가 단순히 구간만 확인하는 것이 아니라, **함수의 모양을 이용해서 다음 점을 예측해 큰 폭으로 조정**하기 때문에 더 빠르다고 추정할 수 있다.

Nonlinear Function

[1, 10] 범위 내에서 직관적으로 결과를 확인하고 싶어 세 개의 근이 모두 [1, 10] 내에 위치하는 삼차함수를 사용하였다.

$$(x - 1) * (x - 5) * (x - 7)$$

Newton with bracketing를 사용해서 Root를 찾은 결과는 아래와 같다.

```
Interval of nonlinear equation where root is likely to exist
Interval [a, b] a: 1.00000, b: 1.00900
Interval [a, b] a: 4.99599, b: 5.00499
Interval [a, b] a: 6.99395, b: 7.00295

-----

Find the roots of nonlinear equation using rtsafe.c
    Root 1 is 1.00000
    Root 2 is 5.00000
    Root 3 is 7.00000
Average convergence speed is 0.0000013333
```

컴파일 결과

```
gcc -o main main.c muller.c ../NRs/ansi/recipes/nrutil.c ../NRs/ansi/recipes/rtbis.c ../NRs/ansi/recipes/rtflsp.c ../NRs/ansi/recipes/rtnwt.c ../NRs/ansi/recipes/rtsafe.c ../NRs/ansi/recipes/rtsec.c ../NRs/ansi/recipes/zbrak.c ../NRs/ansi/recipes/bessj0.c ../NRs/ansi/recipes/bessj1.c -I../NRs/ansi/other -lm
```

- Assignment의 폴더 구조 그대로 사용하였습니다.
- `fmin` 함수가 충돌하는 에러가 발생하여 `nr.h` 에서 `fmin` 관련 함수를 주석 처리하였습니다.