

```
# how to run
$ bash hw2_1.sh
$ bash hw2_2.sh
```

## # hw2\_1 result

```
=====Bisection=====
num of iterations = 17   Root1 = 2.404826
num of iterations = 17   Root2 = 5.520078
num of iterations = 17   Root3 = 8.653728
total_time 0.000027
```

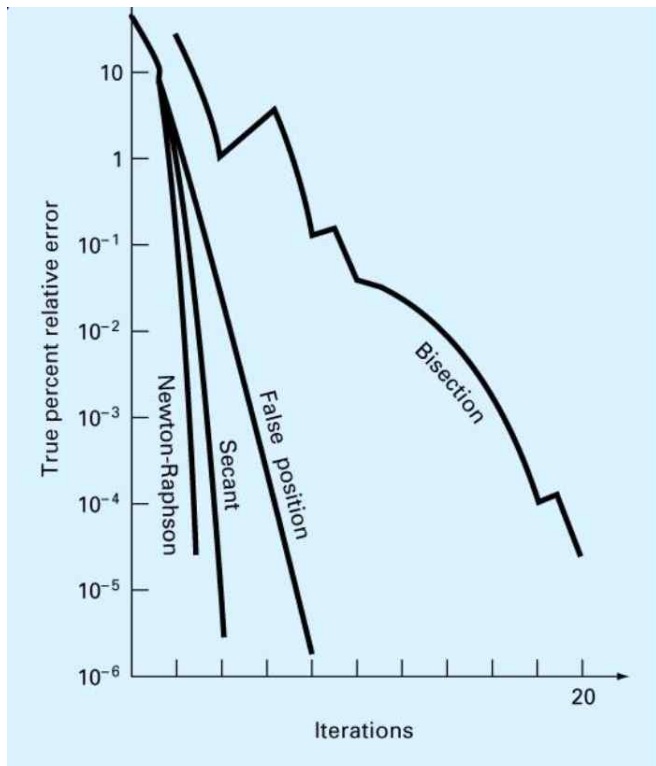
```
=====Linear interpolation=====
num of iterations = 4    Root1 = 2.404826
num of iterations = 3    Root2 = 5.520078
num of iterations = 3    Root3 = 8.653728
total_time 0.000006
```

```
=====Secant=====
num of iterations = 4    Root1 = 2.404825
num of iterations = 3    Root2 = 5.520078
num of iterations = 3    Root3 = 8.653728
total_time 0.000005
```

```
=====Newton-Raphson=====
num of iterations = 3    Root1 = 2.404825
num of iterations = 3    Root2 = 5.520078
num of iterations = 3    Root3 = 8.653728
total_time 0.000005
```

```
=====Newton with bracketing=====
num of iterations = 3    Root1 = 2.404825
num of iterations = 3    Root2 = 5.520078
num of iterations = 3    Root3 = 8.653728
total_time 0.000006
```

Bisection을 제외한 나머지 방식들의 수렴 속도는 위 실험에서는 크게 차이를 보이지 않았다.  
아래 그림을 뒷받침하는 결과인 것 같다.



## # hw2\_2 result

=====Muller=====

Root1 = 2.404825

Root2 = 5.520078

Root3 = 8.653728

num of iterations = 3

total\_time = 0.000017

=====interesting nonlinear equation=====

num of iterations = 3    Root1 = 3.957678

num of iterations = 3    Root2 = 7.086051

total\_time 0.000005

Bisection을 제외한 4가지 방식보다 총 실행시간이 조금 더 걸린다.

nonlinear equation으로 NRs에 있는 bessy 함수를 선택하였다.

bessy 함수는 bessel function Y0다 (그림 참고)

Newton with bracketing으로 구한 답을 그래프와 비교해 보았을때, 올바른 답을 구한 것으로 생각된다.

