

# MAT3008 - Homework 4


박준영

## Contents

1	How to Build	1
2	Problem1	1
3	Problem2	2
4	Pointer to Function in Homework #3	4

## 1 How to Build

다음의 명령어를 수행한다.

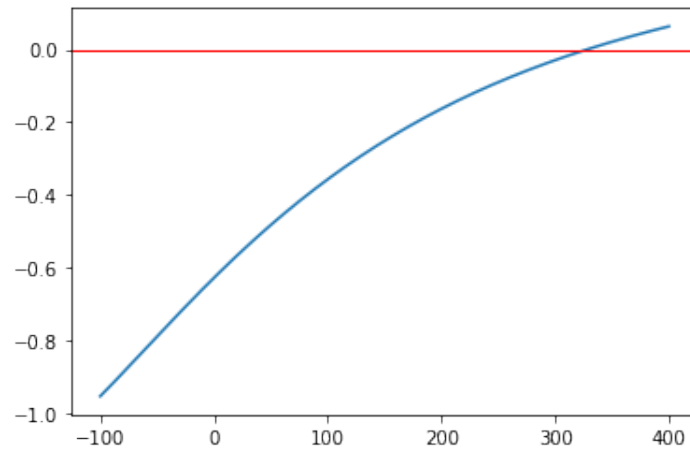
A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. It contains the following text:

```
cd nr  
make  
cd ..  
make
```

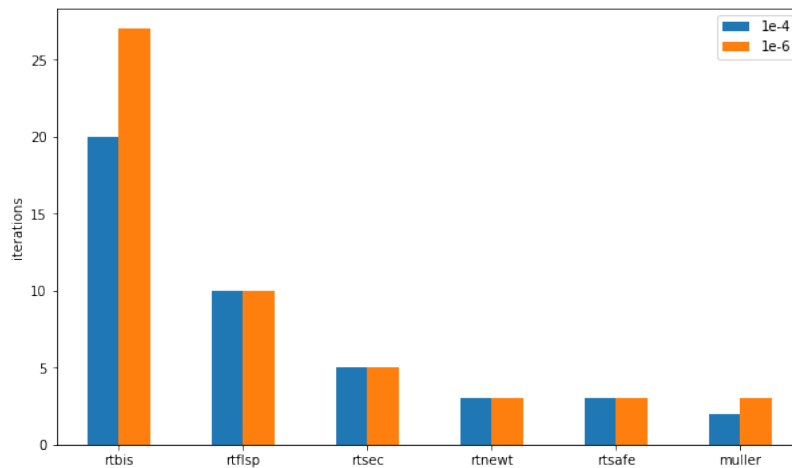
```
cd nr  
make  
cd ..  
make
```

## 2 Problem1

함수  $f(R) = e^{-0.005R} \cos \left[ \sqrt{2000 - 0.01R^2} 0.05 \right] - 0.01$ 의 그래프는 다음과 같다.



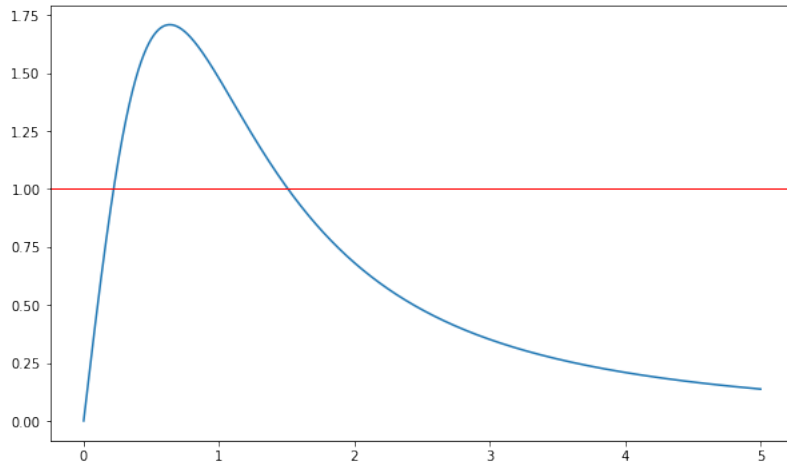
위 그래프를 통해  $f(R) = 0$ 의 해  $R \in [200, 400]$ 인 것을 확인할 수 있다. 따라서  $x_1 = 200, x_2 = 400$ 으로 두고 bisection, linear interpolation, secant, newton-raphson, newton with bracketing, muller methods를 이용하여 해를 구한 결과 bisection method에서  $x_{acc}$ 를  $10^{-4}$ 로 했을 때는  $R = 328.151398$ 이, 위 경우를 제외하면  $R = 328.151328$ 이 나왔다. 각 방법 별 iteration 횟수는 다음의 그래프와 같다.



### 3 Problem2

#8.32

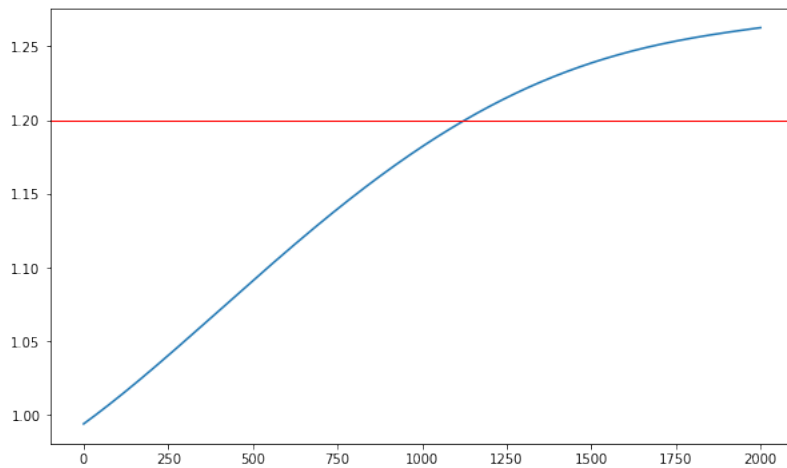
함수  $F(x) = \frac{1}{4\pi\epsilon_0} \frac{qQx}{(x^2+a^2)^{3/2}}$ 의 그래프는 다음과 같다.



위 그래프를 통해  $F(x) = 1$ 의 해는  $[0, 1]$ 에 한 개,  $(1, 2]$ 에 한 개 존재하는 것을 알 수 있다. muller method를 활용하여 구한 해는  $x = 0.221350, 1.509785$ 이다.

### #8.36

함수  $c_p(T) = 0.99403 + 1.671 \times 10^{-4}T + 9.7215 \times 10^{-8}T^2 - 9.5838 \times 10^{-11}T^3 + 1.9520 \times 10^{-14}T^4$ 의 그래프는 다음과 같다.



위 그래프를 통해  $c_p(T) = 1.2$ 의 해는  $x \in [1000, 1250]$ 인 것을 확인할 수 있다. muller method를 활용하여 구한 해는  $T = 1126.0098$ 이다.

## 4 Pointer to Function in Homework #3

함수 포인터는 개념적으로 어떤 함수가 위치한 코드영역의 주소를 가리키는 것이다. 함수 코드 영역의 주소를 알 수 있으면 해당 함수를 호출할 수 있게 된다. 또한 포인터이기 때문에 변수에 저장할 수 있다. 함수 포인터(Pointer to Function)는 다음과 같은 형식으로 사용할 수 있다.

```
<return type> (*<variable name>)(args...)
```

H.W. 3에서는 함수 포인터를 배열에 담아 여러 함수를 for문을 이용해 효과적으로 추가, 호출할 수 있도록 하였다.

```
void* const METHODS[NUM_OF_METHODS] = {
    rtbis,
    rtflsp,
    rtsec,
    rtnewt,
    rtsafe
};
```

위 코드처럼 사용할 method 함수를 배열에 담아두면 아래와 같이 for문을 통해 호출하도록 구현하였다.

```
for (int i = 0; i < NUM_OF_METHODS; ++i)
{
    for (int j = 1; j <= numOfRoots; ++j)
    {
        if (METHOD_TYPES[i] == E_NON_DIFF)
        {
            run(METHODS[i], xb1[j], xb2[j]);
        }
        else
        {
            run_diff_method(METHODS[i], xb1[j], xb2[j]);
        }
    }
}
```