2018008531 송연주

Homework #4

- **Explain the concept of "pointer to function" and describe how you used it in your homework #3**

Numerical Recipes에는 수치해석과 관련하여 편리하게 사용할 수 있는 여러 함수들의 묶음을 제공했다. 이 중 homework #3에 사용된 것은 Bessel함수를 나타내는 `bessj0.c`, `bessj1.c`와 root finding에 사용되는 함수인 `zbrak.c`, `rtbis.c`, `rtflsp.c`, `rtsec.c`, `rtnewt.c`, `rtsafe.c`이다. 이 함수들은 전부 함수 포인터를 인자로 넘겨받아 동작을 한다는 공통점이 있으며 여기에 pointer to function이 사용되었다.

`zbrak.c`, `rtbis.c`, `rtflsp.c`, `rtsec.c`은 float 1개를 매개변수로 받고 float를 return하는 함수를 인자로 받는다. 따라서 그 형식에 부합하는 모든 함수를 각 함수에 넣어 root finding을 하는 데에 사용할 수 있다. `rtnewt.c`, `rtsafe.c`는 float 1개와 float pointer 2개를 매개변수로 받는 함수를 인자로 받는다. 마찬가지로 이 형식에 부합하는 함수라면 모두 이를 이용해 root finding을 할 수 있다.

homework #3에서는 Bessel 함수의 root를 구해야했는데 J0함수인 `bessj0.c`는 float 1개를 받으면 해당 x값에서의 Bessel함수의 값을 구해주는 함수이므로 이를 매개변수로 넣어 Bisection, Linear interpolation, Secant 방법으로 주어진 구간에서의 Bessel함수의 근을 구할 수 있었다.

Newton-Raphson과 Newton with bracketing을 이용하기 위해서는 해당 함수들이 다른 형식의 함수 포인터를 매개변수로 받고 있으므로 `bessj0.c`와 `bessj1.c`를 이용해 새로운 함수를 만들어야 했다. Bessel함수를 미분한 것은 -J1(x)과 같으므로 다음과 같은 함수를 새롭게 정의할 수 있다.

```
void bessj(float x, float* a, float* b)
{
    //...
    *a = bessj0(x);
    *b = -bessj1(x);
}
```

이를 각 함수포인터를 이용해 각 함수의 인자로 넣어 Newton-Raphson과 Newton with bracketing 방법을 이용해 주어진 구간에서의 Bessel함수의 근을 구할 수 있었다.

DATE NO

2018008531 송연주

#8.32

$e_0 = 8.85\times10^{-12} C^2$, $q = Q = 2\times10^{-5} C$, $a=0.9$

$$1 = \frac{1}{4\pi\times 8.85\times10^{-12}\cancel{C^2}} \times \frac{(2\times10^{-5}\cancel{C})^2 x}{(x^2+(0.9)^2)^{\frac{3}{2}}}$$

$$1 = 3.59672 \times \frac{x}{(x^2+(0.9)^2)^{\frac{3}{2}}}$$

$\rightsquigarrow$ $f(x) = 1 - 3.59672\times \dfrac{x}{(x^2+(0.9)^2)^{\frac{3}{2}}}$ 의 근을 구하면 된다.

$\therefore x = 0.22135$ or $x = 1.50978$

#8.36

$1.2 = 0.99403 + 1.671\times10^{-4}T + 9.7215\times10^{-8}T^2 - 9.5838\times10^{-11}T^3$
$+ 1.9520\times10^{-14}T^4$

다음의 식을 만족시키는 T를 구한다.

$\therefore T = -1289.95$ or $T = 1126.01$