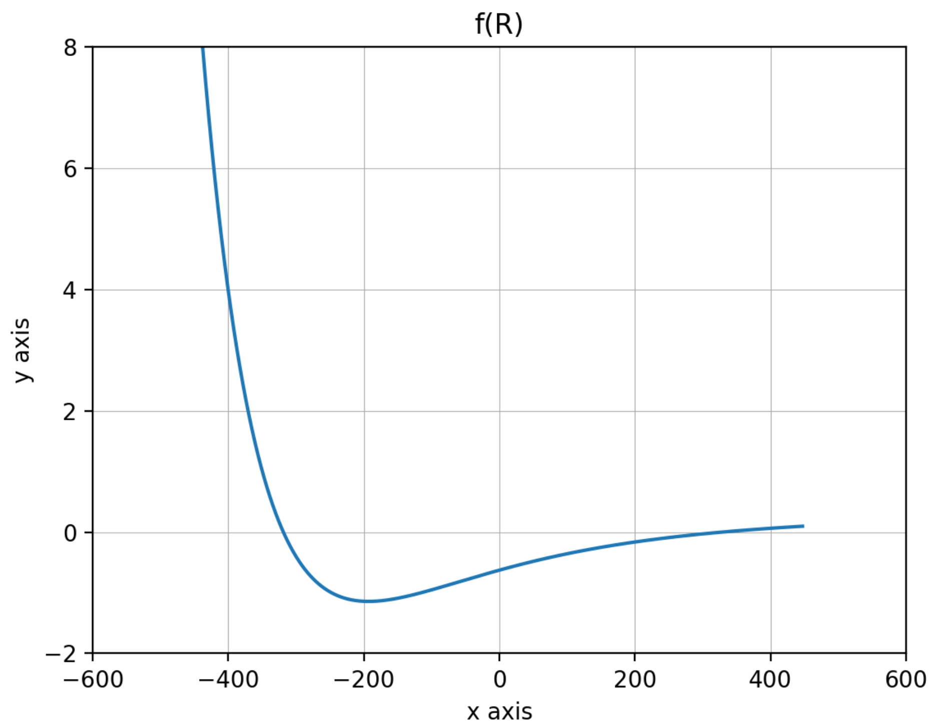
Homework#4

2017029589 컴퓨터소프트웨어학부 류지범

모든 문제의 풀이는 main.cpp에 존재하고, 그래프는 python의 matplotlib로 그렸다.

1. Find the roots of f(R) = 0

해당 식을 그래프로 표현하면 다음과 같다.



그래프를 통해 해는 [0, 400]에 있음을 알 수 있다.

해는 bisection, linear, secant, newton-raphson, newton with bracketing method를 사용해서 구했고, accuracy는 1e-4, 1e-6에 대하여 각각 구했다.

해를 구한 결과는 다음과 같다.

```
Find the root of f(R) = 0
Bisection Method, accuracy 1.0e-4
num of iteration: 21
root: 328.1514168

Linear interpolation, accuracy 1.0e-4
num of iteration: 18
root: 328.1514935

Secant Method, accuracy 1.0e-4
num of iteration: 5
root: 328.1514291

Newton-Raphson Method, accuracy 1.0e-4
num of iteration: 4
root: 328.1514291

Newton with bracketing Method, accuracy 1.0e-4
num of iteration: 4
root: 328.1514291

Bisection Method, accuracy 1.0e-6
num of iteration: 28
root: 328.1514287

Linear interpolation, accuracy 1.0e-6
num of iteration: 24
root: 328.1514299

Secant Method, accuracy 1.0e-6
num of iteration: 6
root: 328.1514291

Newton-Raphson Method, accuracy 1.0e-6
num of iteration: 4
root: 328.1514291

Newton with bracketing Method, accuracy 1.0e-6
num of iteration: 4
root: 328.1514291
```

Iteration의 수를 출력하기 위해서 각 method에서 iteration의 수를 출력하도록 numerical recipe 함수의 코드를 수정했다. 함수가 리턴될 때의 iteration number를 출력하도록 했다.

그 예시는 다음과 같다.

```
if (fabs(dx) < xacc) {
    cout << "num of iteration: " << j << "\n";
    return rtn;
}
```

Bisection과 linear method만 20번대의 iteration을 보여주며 나머지 method는 4~6번 정도로 비교적 빠른 속도로 수렴함을 알 수 있다.

2. Problem 8-32

문제의 식의 형태를 바꾸고 값을 대입해서 정리하면 다음과 같다.

$$F = \frac{1}{4\pi\varepsilon_0}\frac{qQx}{(x+a^2)^{\frac{3}{2}}}$$

$$\Rightarrow \frac{1}{4\pi\varepsilon_0}\frac{qQx}{(x+a^2)^{\frac{3}{2}}} - 1 = 0$$

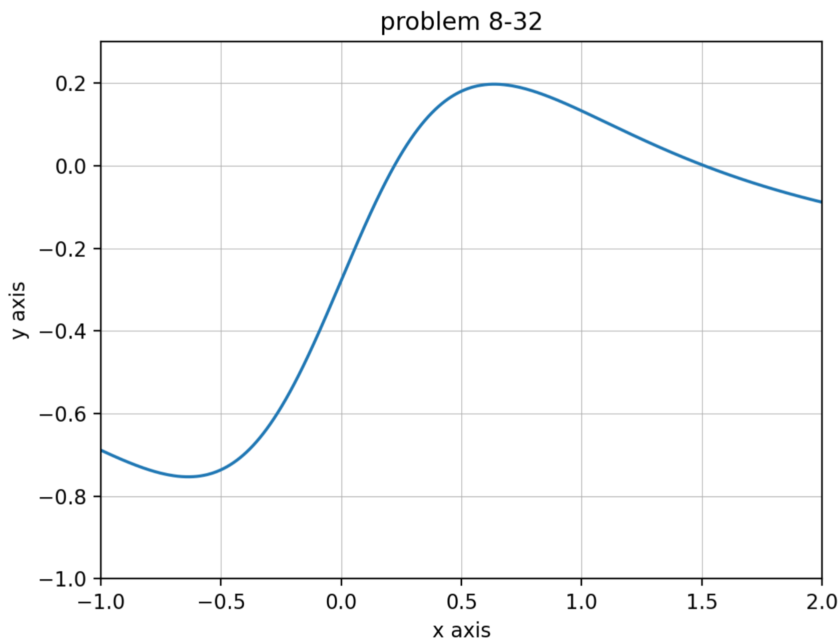
$$\Rightarrow \frac{qQ}{4\pi\varepsilon_0}\cdot\frac{x}{(x+a^2)^{\frac{3}{2}}} - 1 = 0$$

$$\Rightarrow \frac{x}{(x^2+a^2)^{\frac{3}{2}}} - \frac{4\pi\varepsilon_0}{qQ} = 0$$

$q \cdot Q$ 에 2×10^{-5}, $\varepsilon_0 = 8.85\times10^{-12}$, $a = 0.9$ 대입

$$\Rightarrow \frac{x}{(x^2+a^2)^{\frac{3}{2}}} - \frac{8.85\pi}{100} = 0$$

정리한 식에 대해서 그래프를 그려보면 다음과 같다.



해는 [0, 0.5], [1, 2]에 존재함을 알 수 있다.

해당 식과 도함수는 다음과 같이 표현할 수 있다.

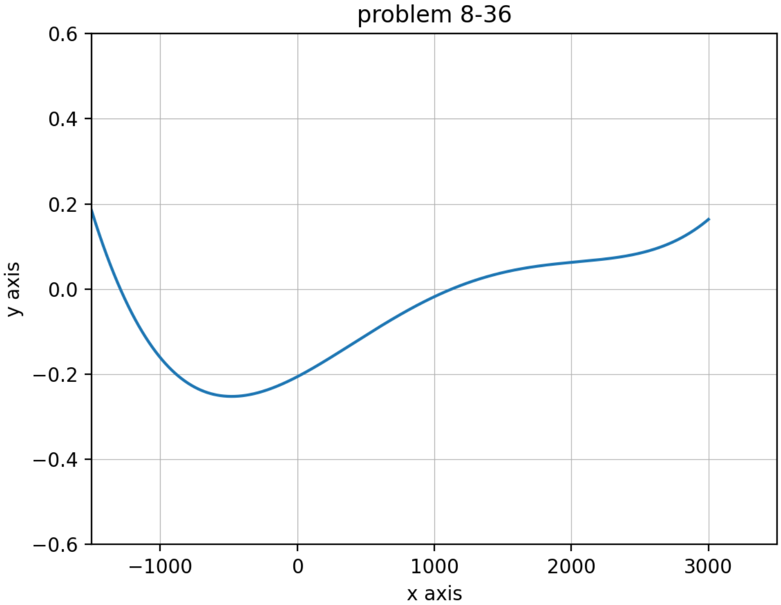
```
DP func1(DP x) {
    return x / pow((x*x + 0.81), 1.5) - 8.85 * M_PI / 100;
}

void funcd1(DP x, DP &f, DP &df) {
    f = func1(x);
    double temp = x*x + 0.81;
    df = 1 / pow(temp, 1.5) - 3 * x*x / pow(temp, 2.5);
}
```

해를 구한 것은 다음과 같다. Accuracy는 1e-4로 설정했다.

```
problem 8-32
Bisection Method in [0.0, 0.5], accuracy 1.0e-4
num of iteration: 12
root: 0.2213134766

Linear interpolation in [0.0, 0.5], accuracy 1.0e-4
num of iteration: 5
root: 0.2213652842

Secant Method in [0.0, 0.5], accuracy 1.0e-4
num of iteration: 5
root: 0.2213501221

Newton-Raphson Method in [0.0, 0.5], accuracy 1.0e-4
num of iteration: 2
root: 0.2213501197

Newton with bracketing Method in [0.0, 0.5], accuracy 1.0e-4
num of iteration: 2
root: 0.2213501197

Bisection Method in [1.0, 2.0], accuracy 1.0e-4
num of iteration: 13
root: 1.50982666

Linear interpolation in [1.0, 2.0], accuracy 1.0e-4
num of iteration: 5
root: 1.509790613

Secant Method in [1.0, 2.0], accuracy 1.0e-4
num of iteration: 4
root: 1.509785138

Newton-Raphson Method in [1.0, 2.0], accuracy 1.0e-4
num of iteration: 1
root: 1.50978514

Newton with bracketing Method in [1.0, 2.0], accuracy 1.0e-4
num of iteration: 1
root: 1.50978514
```

3. Problem 8-36

문제의 식에 대해 그래프를 그려보면 다음과 같다.



식의 해는 [1000, 1500]에 있음을 알 수 있다.

식과 도함수를 표현하면 다음과 같다

```
DP func2(DP x) {
    return 1.9520e-14*pow(x, 4) - 9.5848e-11*pow(x, 3) + 9.7215e-8*x*x + 1.671e-4*x - 0.20597;
}

void funcd2(DP x, DP &f, DP &df) {
    f = func2(x);
    df = 1.9520e-14*pow(x, 3) * 4 - 9.5848e-11*pow(x, 2) * 3 + 9.7215e-8*x * 2 + 1.671e-4;
}
```

해를 구한 것은 다음과 같다. Accuracy는 1e-4로 설정했다.

```
problem 8-36
Bisection Method in [1000.0, 1500.0], accuracy 1.0e-4
num of iteration: 22
root: 1126.11711

Linear interpolation in [1000.0, 1500.0], accuracy 1.0e-4
num of iteration: 6
root: 1126.117164

Secant Method in [1000.0, 1500.0], accuracy 1.0e-4
num of iteration: 4
root: 1126.117161

Newton-Raphson Method in [1000.0, 1500.0], accuracy 1.0e-4
num of iteration: 3
root: 1126.117161

Newton with bracketing Method in [1000.0, 1500.0], accuracy 1.0e-4
num of iteration: 3
root: 1126.117161
```

4. How to use “pointer to function”

Homework#3에선 여러가지 method로 해를 구한다. Method가 하나라면 pointer to function을 굳이 사용할 필요가 없겠지만, 많기 때문에 사용하지 않으면 코드의 중복이 심하게 된다. 하나의 method를 사용할 때마다 for loop를 사용하고 걸리는 시간을 계산하는 과정을 코드로 작성해야하기 때문이다. 이를 한번에 해결하기 위해선 함수가 함수를 호출해야하고 pointer to function이 필요하다.

Function pointer란 함수의 주소를 저장하는 변수이다. 함수의 원형과 type만 알고 있으면 사용할 수 있다.

선언은 다음과 같이 한다.

```
void bessel0(DP (*method)(DP (*func)(DP), DP, DP, DP), DP (*func)(DP), Vec_O_DP xb1, Vec_O_DP xb2,
    int nroot, double xacc) {
    clock_t start, end;

    start = clock();
    for (int i = 0; i < nroot; ++i) {
        double root = (*method)(func, xb1[i], xb2[i], xacc);
        cout << i + 1<< "'th root: " << root << "\n";
    }
    end = clock();
    cout << end - start << "ms passed\n\n";
}
```

호출은 다음과 같이 했다.

```
bessel0(rtbis, func, xb1, xb2, nroot, xacc);
```

함수의 이름을 변수처럼 사용하고, 매개변수는 따로 같이 넘겨준다.