고급 소프트웨어 실습1

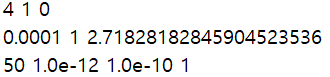
4주차 과제 보고서

20151523

김동현

1. **문제 4**
   1. **Newton-Rahpson Method**

* 입력



Line 1 : 4 = PROBLEM4 / 1 = 풀이방법:Newton-Rahpson / 0 = parameter 개수

Line 2: 0.0001 = 초기값x / 1 = Convergence check / 2.7182….. = 이미 알고있는 근

Line 3 : 50 = 최대 Iteration / 1.0e-12 = delta / 1.0e-10 = epsilon / 1 = 중간과정

* 출력

텍스트, 표시중, 사람이(가) 표시된 사진

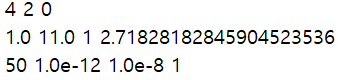
자동 생성된 설명

우선 iteration을 10개로 증가시키기 위해 초기값 x를 x > 0 이면서 아주 작은 값으로 정하였다.

해의 오차가 Order of e-15보다 작은 값이 되어버리면 일치한다고 간주되어서 해 x 와 자연 상수 e는 오차가 있지만, 해의 오차는 없다고 출력이 되었다. 그래서 delta 값을 1.0e-15보다 큰 1.0e-12로 주었다. 그 결과 해의 오차는 약 6.328e-14 정도로 매우 근사함을 볼 수 있다.

그리고 위의 결과로 볼 때, 대략 정도의 rate로 근에 수렴함을 알 수 있다.

* 1. **Secant Method**
* 입력



Line 1 : 4 = PROBLEM4 / 2 = 풀이방법: Secant / 0 = parameter 개수

Line 2 : 1.0 = 초기값x0 / 11.0 = 초기값x1

1 = Convergence check / 2.7182….. = 이미 알고있는 근

Line 3 : 50 = 최대 Iteration / 1.0e-12 = delta / 1.0e-8 = epsilon / 1 = 중간과정

* 출력

텍스트, 표시중이(가) 표시된 사진

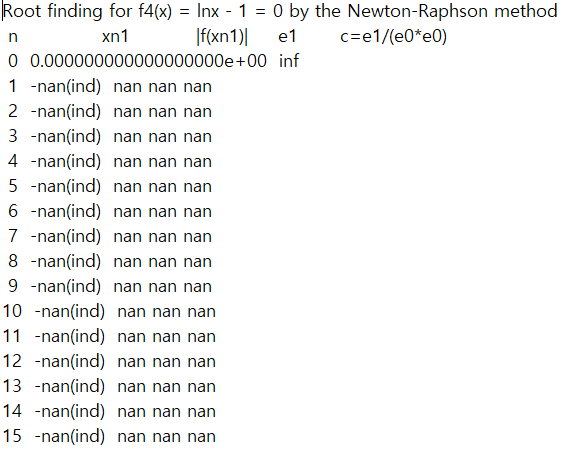
자동 생성된 설명

우선 iteration을 10개로 증가시키기 위해 초기값 x1을 자연 상수 e 보다 큰 값에서 조금씩 증가시키며 조절하였다.

해의 오차가 Order of e-15보다 작은 값이 되어버리면 일치한다고 간주되어서 해 x 와 자연 상수 e는 오차가 있지만, 해의 오차는 없다고 출력이 되었다. 그래서 delta 값을 1.0e-15보다 큰 1.0e-12로 주었다. 그 결과 해의 오차는 약 1.33e-15 정도로 아슬아슬하게 1.0e-15보다 커서 관측 가능한 범위 내에서는 매우 근사한 결과값을 얻을 수 있었다.

그리고 위의 결과로 볼 때, 대략 정도의 rate로 근에 수렴함을 볼 수 있다.

* **Newton-Rahpson과 Secant Method의 발산의 경우**



위의 경우 초기값에 의해 발산이 일어난 경우이다.

초기값 x0가 대략 x0 >= 7.3891 || x0 <= 0 의 범위에 있을 때 발산이 일어난다.

라켓볼, 개체이(가) 표시된 사진

자동 생성된 설명

위의 그래프는 y = lnx – 1를 나타낸 것이다.

X가 약 자연상수 e에 있을 때 y=0을 지나는 모습을 볼 수 있다. 그렇다면 x0가 7.3891..인 지점이 의미하는 것은 무엇일까?

바로 ln(7.3891….) = 2 인 것이다.

Newton-Rahpson의 계산식 𝑥𝑛+1 = 𝑥𝑛 − 𝑓(𝑥𝑛) /𝑓′(𝑥𝑛) (Secant의 경우는 𝑥𝑛+1 = 𝑥𝑛 − 𝑓(𝑥𝑛) \*( 𝑥𝑛 − 𝑥𝑛−1 ) / (𝑓 𝑥𝑛 − 𝑓(𝑥𝑛−1))) 에 f(x) = lnx-1, n=0을 대입해 보면

X1 = 7.3891.. – ln(7.3891..)/(1/7.3891) = 0이 된다.

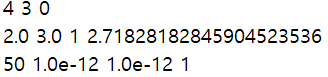
만약 Xn이 7.3891.. 보다 커지게 되면 Xn+1은 0보다 작은 값이 되어진다.

하지만 로그 연산의 정의에 의해 lnX에서 X는 X>0이어야 정의되기 때문에 위와 같이 발산이 일어남을 알 수 있다.



위의 그림은 x=7.3891..일 때 ln(x) - 1 = 1임을 보여주는 그래프이다.

* 1. **Bisection Method**
* 입력



Line 1 : 4 = PROBLEM4 / 3 = 풀이방법: Secant / 0 = parameter 개수

Line 2 : 2.0 = 초기값x0 / 3.0 = 초기값x1

1 = Convergence check / 2.7182….. = 이미 알고있는 근

Line 3 : 50 = 최대 Iteration / 1.0e-12 = delta / 1.0e-12 = epsilon / 1 = 중간과정

* 출력

텍스트이(가) 표시된 사진

자동 생성된 설명

Bisection Method에서는 Iteration이 10개 이상이 되는 경우가 많으므로, 적당히 초기값 x0와 x1을 설정하였다.

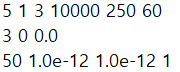
해의 오차가 Order of e-15보다 작은 값이 되어버리면 일치한다고 간주되어서 해 x 와 자연 상수 e는 오차가 있지만, 해의 오차는 없다고 출력이 되었다. 그래서 delta 값을 1.0e-15보다 큰 1.0e-12로 주었다. 그 결과 해의 오차는 약 9.79e-13 정도로 매우 근사함을 볼 수 있다.

그리고 위의 결과로 볼 때, 대략 정도의 rate로 근에 수렴함을 볼 수 있다.

1. **문제 5**

**2-1. 𝑊 = 10000, 𝑀 = 250, 𝑘 = 60일 경우**

* 입력



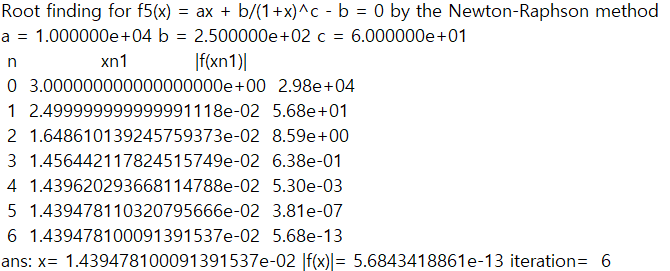
Line 1 : 5 = PROBLEM5 / 1 = 풀이방법:Newton-Rahpson / 3 = parameter 개수

10000 = W / 250 = M/ 60 = k

Line 2: 3 = 초기값x / 0 = Convergence check X / 0.0

Line 3 : 50 = 최대 Iteration / 1.0e-12 = delta / 1.0e-12 = epsilon / 1 = 중간과정

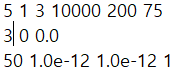
* 출력



X의 추정 범위를 0<x<1로 가정하여서 초기값으로 3을 넣었다. 그 결과, 해의 오차는 약 5.684 e-13으로 근사한 값을 얻을 수 있었다.

**2-2. 𝑊 = 10000, 𝑀 = 200, 𝑘 = 75일 경우**

* 입력

****

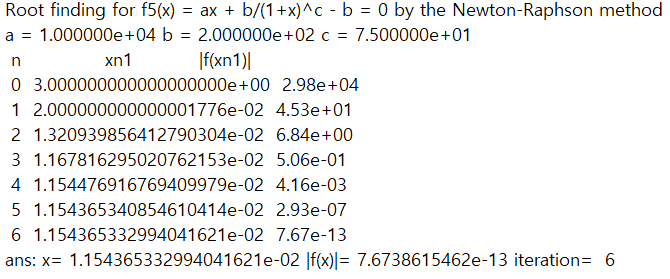
Line 1 : 5 = PROBLEM5 / 1 = 풀이방법:Newton-Rahpson / 3 = parameter 개수

10000 = W / 200 = M/ 75 = k

Line 2: 3 = 초기값x / 0 = Convergence check X / 0.0

Line 3 : 50 = 최대 Iteration / 1.0e-12 = delta / 1.0e-12 = epsilon / 1 = 중간과정

* 출력



X의 추정 범위를 0<x<1로 가정하여서 초기값으로 3을 넣었다. 그 결과, 해의 오차는 약 7.674 e-13으로 근사한 값을 얻을 수 있었다.

**2-3. 결론**

F(x) = ax + b/(1+x)^c – b = 0/ (a = W, b = M, c = k)

로 두고 계산하였다.

그 결과 (M, k) = (250 , 60) 인 경우에는 이자율 r = 1.4395 이고, (200, 75) 인 경우에는

이자율 r = 1.1547이다. (각 r값은 소수점 아래 다섯째 자리에서 반올림 실시한 값이다.)

즉, 채무자의 입장인 길동에 있어서는 이자율이 낮은 편이 유리하므로 이자율이 더 낮은 2번째식(W = 10000, M = 200, k = 75)의 상환이 유리하다.