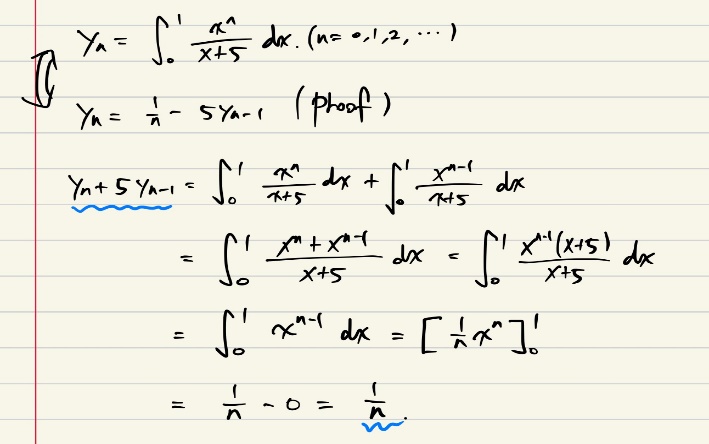
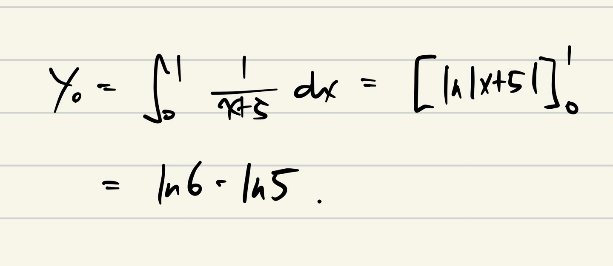
**Problem Set #1**

Numerical analysis

201621505 채진기

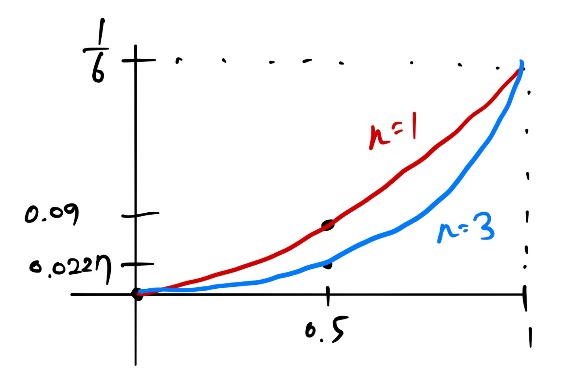
**1 – (a).** Yn에 대한 점화식 증명입니다. **(b)** Y0 값은 다음과 같습니다.



**(c)** 우선, Yn의 값을 생각해보면 구간 [0,1]에서의 적분 값입니다. 피적분함수는 X^(n)/x+5 로서

구간내에서 양수 값 만을 갖습니다. 따라서 양수 값 만을 갖는 함수의 적분 값은 0보다 클 수

밖에 없고 (Yn>0) 함수의 degree , n 이 커짐에 따라 함수의 그래프는 가파른 곡선을 보여줍니다.

직관적으로 알 수 있지만 n에 값을 조금 바꿔서 그래프

를 그려보면 n값이 커질수록 더욱 가파른 것을 알 수

있고, 이는 적분 값의 감소를 의미합니다. (Yn<Yn-1)

반복적으로 n의 값을 높여서 0.9에서의 함수값을 계산

해보면 다음과 같습니다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | 0 | 1 | 2 | 3 | … | 12 | 13 | 14 | 15 |
| f(x) | 0.169492 | 0.152542 | 0.137288 | 0.123559 | … | 0.047869 | 0.043082 | 0.038774 | 0.034897 |

0.9에서의 함숫값이 작아진 만큼 그 적분값은 작아질 것이고, 따라서 Yn의 값 또한 작아집니다.

그래서 n이 무한대로 수렴할 때, Yn또한 0으로 수렴하는 관계를 가집니다. (Yn -> 0 as n -> inf)

**(d)** n=4 까지의 Yn을 계산해보면 다음과 같습니다. (소수점 4자리에서 반올림한 값)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n | 0 | 1 | 2 | 3 | 4 |
| Yn | 0.182 | 0.09 | 0.05 | 0.833 | -0.166 |

최초 y0의 값을 정확도 10^(-3)의 수준으로 정의하니 Yn의 값이 n이 커짐에 따라 커지거나,

음수가 나와버리는 오류가 발생했습니다. 이는 매우 부정확한 iteration 결과입니다.

**(e)** 소스 코드와 그것을 실행한 결과물을 첨부했습니다. (메모장, 엑셀, png파일)

결과를 살펴보면, y20 값부터 비정상적으로 작동하는 것을 볼 수 있습니다.

이론상으로는, y의 값이 ln6 – ln5 부터 작아지면서 0으로 수렴하는 꼴이어야 하는데,

Y20의 값은 y19보다 크며, 심지어 y21은 음의 값을 나타냅니다.

**(f)** 소스 코드와 그것을 실행한 결과물을 첨부했습니다. (메모장, 엑셀, png파일)

(e) 번 문제와는 다르게 (y49를 제외하고) 오류 없이 작동하는 것을 볼 수 있습니다.

n이 작아지면서 y의 값이 커지는 것을 볼 수 있습니다. 물론 음의 값 또한 발견되지 않았습니다.

**2.** 소스 코드와 그것을 실행한 결과물을 첨부했습니다. (메모장, png파일)

Stopping criterion 은 발견한 x에서의 원함수의 해의 절댓값이 0.001 보다 작을 때로 정했습니다.

어느정도 예상되는 지점에 초기 3개의 점을 넣어주니, 빠른 속도로 해를 근사하는 것을

볼 수 있었습니다. 하지만 최초 3개의 점을 1, 2, 3 과 같이 무작위로 지정해주니 프로그램이

잘 작동하지 않았습니다.

**3 - (a).** 소스 코드와 그것을 실행한 결과물을 첨부했습니다. (메모장, 엑셀, png파일)

그래프는 첨부한 엑셀파일을 통해 표현했습니다.

**3- (b).** 소스 코드와 그것을 실행한 결과물을 첨부했습니다.(메모장, 엑셀, png파일)

두 가지 방식으로 같은 함수 f(x) = 1 / (1 + 25x^(2)) 에 대해 근사 해보았습니다.

라그랑주 방식으로는 양 끝쪽을 제외하고 꽤나 잘 근사 했으나..

큐빅 스플라인 방식은 제가 식을 잘못 써준 탓인지 초반에는 잘 근사하다가 우측 끝에서

근사 결과에 크게 오차가 발생했습니다.

제 코딩 실수를 제외하면 큐빅 스플라인 방식이 더 정확하게 근사하는 것을 보았습니다.

(엑셀파일에 결과와 그 형태를 표현했습니다.)