**수치해석 #1 assignment**

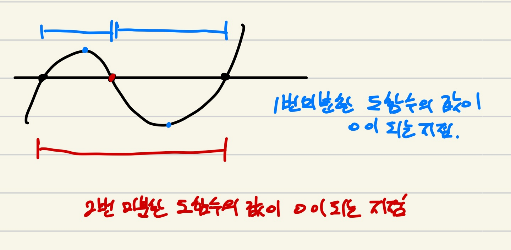
**201621505 채진기**

**1. (Slide 26) Prove the Generalized Rolle’s Theorem. (Giving an idea is enough.)**

F(x) = 0 이 되는 지점이 n+1번 존재한다면, n개의 구간으로 나눠 볼 수 있습니다.

각 구간에서 한번 미분한 도함수F’(x)의 값이 0이 되는 지점이 1개 이상 존재함을 알 수 있습니다.

같은 맥락에서 두개의 구간 내에 F’’(x) = 0 되는 지점이 1개 이상 존재함을 알 수 있고,

따라서 n번 미분한 도함수의 값이 0이 되는 지점이 존재합니다.

(3차함수를 예로 들었습니다.)

**2. Representable numbers**

**(a) (Slide 39) Are all rational numbers representable exactly by 64-bit representation?**

**Why or why not?**

표현할 수 있는 가장 작은 숫자, 약 0.22251\*10^(-307) 보다 작은 수는 Underflow 처리되고,

반대로 약 0.17977\*10^(309) 보다 큰 수는 Overflow 처리되기 때문에

저 범위에 속해 있지 않은 유리수는 포함되지 않습니다.

그리고 소수점 자릿수가 52자리 이상 넘어가는 유리수 또한 표현되지 못합니다.

**(b) (Slide 39) If you check in MATLAB, logical value of the statement 0*.*95*−* 0*.*94 == 0*.*01**

**is ‘false.’ Explain why**

0.95라는 수는 컴퓨터에서 0.111’1001’’1001’…(2) 로 저장되고 정확하게 표현할 수 없는 수입니다.

마치 10진법에서 1/3이 10의 거듭제곱수로 표현될 수 없는 것과 같은 맥락입니다.

따라서 컴퓨터가 0.95와 0.94를 각각 저장한 후 연산할 때 그 결과가 0.01보다 더 가까운

다른 근사값이 존재하기 때문에 ‘false’ 라고 출력됩니다.

**(c) (Slide 39) Can two diﬀerent 64-bit representation have same nonzero value?**

**Why or why not?**

64비트에서 어떤 숫자를 표현할 때 s, c, f 세 영역으로 나누어 저장하는데,

f 영역(mantissa), 1+f 로 결정되는 영역은 1이상 2미만의 값을 갖고

c 영역(characteristic), 2^(C-1023)로 결정되는 영역은 2의 지수 값을 가지므로

두 영역이 서로 영향을 줄 수 없어 한 64비트 표현이 유일한 한가지 수를 지정합니다.

**(d) (Slide 41) 2*−*100 is representable and (11*.*11*· · ·* 1)2 is not representable by 64-bit repre-**

**60 1*s***

**sentation, even though the former is much smaller than the latter. Explain why.**

전자는 64비트 c 영역(characteristic)에서 스케일을 바꿔 2의 지수 값으로 표현될 수 있지만,

소수점은 52자리까지만을 표현할 수 있기 때문에 전자보다 더 큰 값이라도 표현할 수 없습니다.

**(e) (Slide 41) Suppose you found that 2*−*100 is not representable and (11*.*11*· · ·* 1)2 is repre-**

**~60 1*s~***

**sentable, by an old computer. Can you guess why?**

스케일을 바꿀 수 없는 오래된 컴퓨터는 고정소수점 방식으로 정수, 소수를 표현할 비트 수를

정해 놓고 숫자를 표현하는데, 소수를 표현하는 비트를 늘려 (11*.*11*· · ·* 1)(2)를 표현했을 것입니다

그리고 2^(-100)은 그보다 많은 소수점 자릿수를 갖기 때문에 표현할 수 없었을 것이라고

추측합니다.

**3. Reducing round-oﬀ error**

1. **(Slide 47) The eﬀect of the reformulation is diﬀerent for two roots. Explain why.**

Reformulation의 과정 중 유리화를 통해 연산을 진행하는데,

그 과정에서 비슷한 숫자를 빼는 연산이 다른 root에 포함되어 한 root는 정확한 값이 도출됐고

또 다른 한 root는 오히려 부정확한 값이 도출되었습니다.

비슷한 숫자를 빼는 연산을 가진 root가 큰 오차를 갖는 값을 갖습니다.

**(b) (Slide 48) Solve the problem in two diﬀerent ways and check relative errors.**

Solve [x^(3)-6.1x^(2)+3.2x+1.5 at x=4.71]

when not using 3digit arithmetic) 104.487111 - 135.32301 + 15.072 + 1.5 = -14.263899 = A

when using 3digit arithmetic) 105-135+15.1+1.5 = -13.4 = B

relative error is |A – B| / |A| = 0.06

Solve [((x-6.1x)x + 3.2)x + 1.5 at x=4.71]

when not using 3digit arithmetic) 104.487111 - 135.32301 + 15.072 + 1.5 = -14.263899 = A

when using 3digit arithmetic) (중간 수식계산 생략) -14.3 = C

relative error is |A – C| / |A| = 0.0025