1. 수학적 기초 및 오차 해석

Mathematical Preliminaries and Error Analysis

1. 수학적 기초 (Mathematical basis)

- Taylor Series: $f(x)=\sum_{k=0}^{\infty}\frac{f^{k}(x_o)}{k!}(x-x_o)^k$

2. 오차 (Error)

2.1 측정오차 (Measurement error)

- 오차의 원천 (source of error)
  - 사람, 측정기, 측정방법, 환경 (human, device, method, environment)
- 측정 오차 줄이기 (to reduce the error)
  - 반복 (repeat)
  - 인자들의 조합(실험 계획) (design of experiment)
  - 통계 처리 기술 (statistical treatment)
- 상대오차의 개념 (concept of relative error)

2.2 유효숫자 (significant figure; round-off error)

$$\left(\sqrt{3}\right)^2=3 \qquad \sqrt{3}=1.7321 \qquad 1.7321^2=3.0002$$

- Computer 의 숫자 처리(32 bit): 1(부호), 7(지수), 24(숫자)
- Computer 의 숫자 처리(64 bit): 1(부호), 11(지수), 52(숫자)
- 자리 수 올리기: Rounding/Chopping

2.3 근사오차 (truncation error)

$$f'(a)\approx\frac{f(b)-f(a)}{b-a}$$

$$f(b)=f(a)+f'(a)(b-a)+\frac{f''(a)}{2!}(b-a)^2+\cdots$$

$$f'(a)=\frac{f(b)-f(a)}{b-a}-\frac{f''(a)}{2!}(b-a)+\cdots$$

2.4 오차의 전파 (propagation of error)

$$x=x_0\pm\Delta x,\ y=y_0\pm\Delta y$$

$$xy=(x_0\pm\Delta x)(y_0\pm\Delta y)=x_0y_0\pm x_0\Delta y\pm y_0\Delta x\pm\Delta x\Delta y$$

Appendix 1. Computer 에서의 수의 표현 (representation of numbers)

- 컴퓨터에서의 정수, 실수 표현 시연
- Computer 에서 수의 체계 (number system): $(-1)^s \times 2^{c-1023} \times (1 + f)$

  s: sign indicator (the first bit)

  c: exponent (11-bit)

  f: mantissa (52-bit)

가수와 exponent 의 bit 수가 수의 범위 및 정밀도에 어떠한 영향을 주는가?
(Effects of the number of bits on the range and accuracy of numbers?)

Example)

0 10000000011 1011100100010000000000000000000000000000000000000000000000

$= + 2^{(1027-1023)} (1+ (1/2 + 1/8 + 1/16 + 1/32 + 1/256 + 1/4096))$

= +27.56640625

Example-2) 위의 숫자와 가장 가까운 크고, 작은 숫자 (neighboring No.)

0 10000000011 1011100100010000000000000000000000000000000000000000000001

= +27.56640625000000022204460***

0 10000000011 1011100100001111111111111111111111111111111111111111111111

= +27.56640624999999977795540***

2.2×10^{-16} 이하 scale 에서는 chopping 또는 rounding
(in scale smaller than 2.2×10^{-16}: chopping or rounding)

Example-3) 컴퓨터에서 표현할 수 있는 가장 크고, 작은 숫자
(biggest and smallest number in computer)

$2^{1023} (1+ (1 - 2^{-52})) \approx 0.17977 \times 10^{309}$

$2^{-1022} (1+ 0) \approx 0.2225 \times 10^{-307}$

Appendix 2. Computer 에서의 산술 오차

- 큰 수와 작은 수 간의 덧셈, 뺄셈으로 인한 유효 숫자의 상실
  (loss of significant figures between large and small numbers)
  Example) 유효 숫자 4 자리에서 (with four significant figures)
  12340+56, 100000-9

- 비슷한 크기의 수 간의 뺄셈으로 인한 유효 숫자의 상실
  (loss of significant figures in subtraction between similar numbers)
  Example) $x^2 + 62.10x + 1 = 0$ (x_1 = -0.01610723, x_2=-62.08390)
  유효 숫자 4 자리에서 x_1 을 구할 경우 -0.02
  비슷한 숫자 간 뺄셈을 피함으로써 error 감소

개인 과제물 (HW#1-1: 예습)

- 0.00001 을 백만 번 더하면서 매 십만 번째마다 결과를 출력하시오.
  (add 0.00001, million times, showing the result every 100,000th step)
- 1 을 백만 번 더하면서 매 십만 번째마다 결과/100000 을 출력하시오.
  (add 1, million times, showing the result/100000 every 100,000th step)

  - Practice writing a code, generating an execution file, and run it
  - Integer vs. real
  - Single precision vs. double precision

개인 과제물 (HW#1-2: 복습)

- 현재 사용하고 있는 컴퓨터가 single precision 에서 구분할 수 있는 수의 정밀도가 어느 정도인지 프로그램을 짜서 확인하고, 이 system 이 몇 개의 bit 로 가수(mantissa)를 표현하는지 판정하시오.
  (estimate the number system (bit) of your computer)