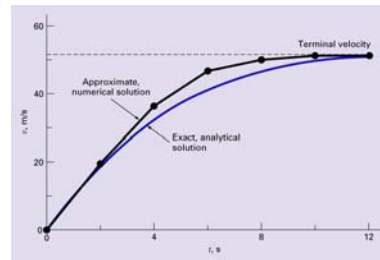


Chap. 4 오차 (Errors)

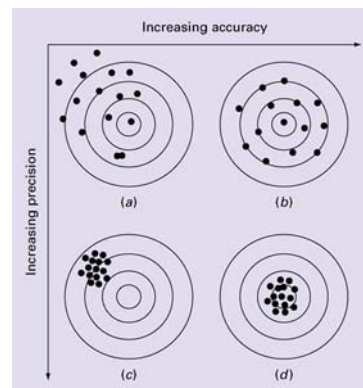
- ❖ 오차의 정의
- ❖ 반올림오차
- ❖ 절단오차
- ❖ 수치미분



오차 (Error)

□ 정확도와 정밀도

- 정확도 (Accuracy):
- 정밀도 (Precision):
- 부정확성 (Inaccuracy, 편심):
- 비정밀도 (Imprecision, 불확실성):
- Ex) p. 67 그림 4-1 (a) 부정확과 비정밀
(b) 정확과 비정밀
(c) 부정확과 정밀
(d) 정확과 정밀



오차 (Error)

□ 오차의 정의

● 절대오차 (absolute error) : (A: 참값, a: 근사값)

● 상대오차 (relative error) :

● 백분율 상대오차 (percent relative error) :

● Ex) 다리와 못의 길이 측정

⇒ 다리: 참값 10000 cm, 측정값 9999 cm →

⇒ 못 : 참값 10 cm, 측정값 9 cm → %



□ 수치해석(반복계산)에서의 상대오차

● 실제 상황에서 참값을 알고 있는지? → 참값을 가장 잘 나타내는 근사값 사용

● 반복계산에서의 상대오차:

● Ex) Exponential 함수의 Maclaurin 급수전개: $e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$

오차의 종류

□ 반올림오차 (Round-off error)

● 디지털 컴퓨터가 어떤 수량을 완전하게 표현할 수 없기 때문에 발생

● 계산의 각 단계에서 유효숫자를 처리해가는 과정에서 발생

● Ex) C 언어에서의 부동소수점 처리

⇒ float 형 (4 byte) :

⇒ double 형 (8 byte) :

□ 절단오차 (Truncation error)

● 실제 계산식을 근사식으로

묘사할 때 발생하는 오차

● 무한급수 혹은 극한조작으로 계산되는 경우에 발생 예) Taylor series

반올림오차 (Round-off error)

□ 유효숫자

5/3을 소수 5자리에서 반올림하면 : 1.6667

- 반올림해서 1.6667이 되는 수 x 의 범위 :
- 오차 :
- Cf) Data 처리시 유효숫자의 중요성: $A = 0.25$, $a = 0.24$ 일 때 오차 0.01

(소수점 2자리에서 반올림) $A = 0.3$, $a = 0.2$ 일 때 오차: 0.1

□ 컴퓨터(32비트)에서의 유효숫자 표현

- 정수: $-2,147,483,648 \sim 2,147,483,647$ ($-2^{31} \sim 2^{31}$)
- 소수: $10^{-308} \sim 10^{308}$ (16 bit: $10^{-39} \sim 10^{39}$) cf) **realmin & realmax**
- 기계입실론 (Machine epsilon) : 가장 미세한 숫자 (유효숫자 16자리) cf) **eps**

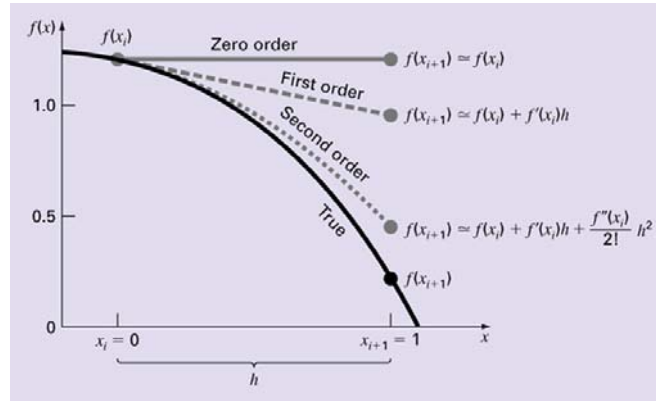
반올림오차

□ 산술적 연산에 의한 오차

- 연산시 유효숫자의 변경에 의해 발생하는 오차 (예: 유효숫자 4개)
 - ⇒ $1.557 + 0.04341 = 0.1557 \times 10^1 + 0.004341 \times 10^1 = 0.160041 \times 10^1 = 0.1600 \times 10^1$
- 비슷한 두 수간의 뺄셈 수행시 유효숫자 감소
 - ⇒ $0.12345 - 0.12343 = 0.00002$
- Ex) $A = 222.88$, $a = 222.11$ $E_t = 222.88 - 222.11 = 0.77$
위 수치를 유효숫자 4자리로 절단하여 E 를 구해보면

절단오차 (Truncation Errors)

Taylor Theorem: Any smooth function can be approximated as a polynomial.



Taylor Series

- First order Taylor approximation of $f(x)$:

- Second order Taylor approximation of $f(x)$:

- n-th order Taylor approximation of $f(x)$:

□ 번지점프 속도 계산: 1st order approximation

truncation error

$$\begin{aligned} \frac{R_1}{t_{i+1} - t_i} &= \frac{1}{t_{i+1} - t_i} \frac{v''(\xi)}{2!} (t_{i+1} - t_i)^2 \\ &= \frac{v''(\xi)}{2!} (t_{i+1} - t_i) \\ &= O(h) \end{aligned}$$

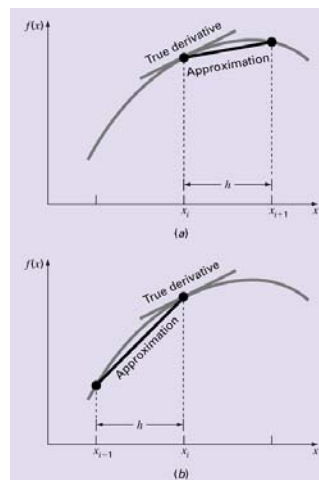


Ex) 예제 4.3 Taylor 급수 전개 (pp. 110)

수치미분 (Numerical Differentiation)

□ 유한차분(Finite Difference)에 의한 수치미분

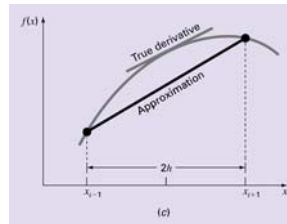
- 전향차분 (forward difference)
- 후향차분 (backward difference)



수치미분 (Numerical Differentiation)

□ 유한차분(Finite Difference)에 의한 수치미분

- 중심 차분 (central difference)



Ex) 예제 4.4 도함수의 유한제차분 근사 (pp. 116 ~ 117)

전체 수치오차의 고찰

□ Step size(h)를 줄이면?

- Taylor Series의 계산이 정확해짐에 따라 절단오차 감소됨
- 반복계산 회수가 증가됨에 따라 반올림오차의 누적량이 많아짐

