

# Approximations & Round-off Errors

**Chapra: Chapter-3**

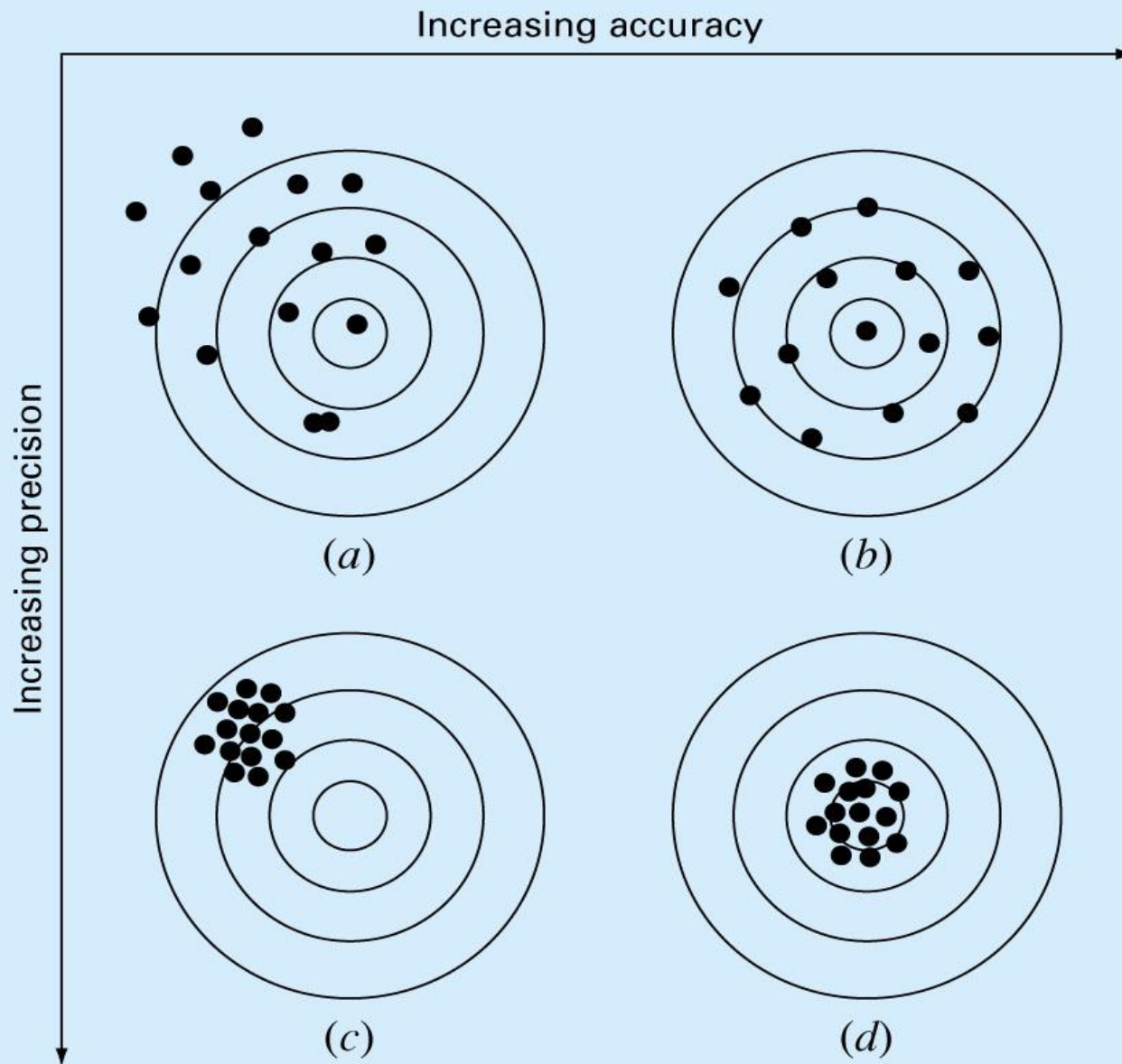


- For many engineering problems, we cannot obtain analytical solutions
- Numerical methods yield approximate solution that are close to the analytical solution. We cannot exactly compute the errors associated with numerical methods.
  - Only rarely given data are exact, since they originate from measurements. Therefore there is probably error in the input information.
  - Algorithm itself usually introduces errors as well, e.g., unavoidable round-offs, etc ...
  - The output information will then contain error from both of these sources.
- How confident we are in our approximate result?
- The question is *“how much error is present in our calculation and is it tolerable?”*



- **Accuracy.** How close is a computed or measured value to the true value
- **Precision (or *reproducibility*).** How close is a computed or measured value to previously computed or measured values.
- **Inaccuracy (or *bias*).** A systematic deviation from the actual value.
- **Imprecision (or *uncertainty*).** Magnitude of scatter.





# Significant Figures

- Number of significant figures indicates precision. Significant digits of a number are those that can be *used* with *confidence*, e.g., the number of certain digits plus one estimated digit.
- Let the speed reading a motorcycle be in between 48 and 49 km/h. Someone says it is 48.8 and another says 48.9 km/h. Two digits 48 are certain and one digit must be estimated.

53,800 How many significant figures?

5.38 x 10<sup>4</sup>                      3

5.380 x 10<sup>4</sup>                     4

5.3800 x 10<sup>4</sup>                  5



# Error Definitions

Numerical errors arise from the use of approximations to represent exact mathematical operations and quantities.

These include


1. *truncation errors*, which result when approximations are used to represent exact mathematical procedures
2. *round-off errors*, which result when numbers having limited significant figures are used to represent exact numbers.



# Error Definitions

True Value = Approximation + Error

$$E_t = \text{True value} - \text{Approximation (+/-)}$$

 True error

$$\text{True fractional relative error} = \frac{\text{true error}}{\text{true value}}$$

$$\text{True percent relative error, } \varepsilon_t = \frac{\text{true error}}{\text{true value}} \times 100\%$$



- For numerical methods, the true value will be known only when we deal with functions that can be solved analytically (simple systems). In real world applications, we will obviously not know the true answer a priori. Then

$$\varepsilon_a = \frac{\text{Approximate error}}{\text{Approximation}} \times 100\%$$

- Iterative approach*, example Newton's method

$$\varepsilon_a = \frac{\text{Current approximation} - \text{Previous approximation}}{\text{Current approximation}} \times 100\%$$

(+ / -)





- Use absolute value.
- Computations are repeated until stopping criterion is satisfied.

$$|\varepsilon_a| < \varepsilon_s$$

Pre-specified % tolerance based on the knowledge of your solution

- If the following criterion is met

$$\varepsilon_s = (0.5 \times 10^{(2-n)})\%$$

you can be sure that the result is correct to at least  $n$  *significant* figures.

# Example 3.2

- In mathematics, function can often be represented by infinite series. For example, the exponential function can be computed using the *Maclaurin series expansion* as:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots$$

Thus, as more terms are added in sequence, the approximation becomes a better and better estimate.

Starting with the simplest version,  $e^x=1$ , add terms one at a time to estimate  $e^{0.5}$ . After each new term is added, compute the true and approximate percent relative errors. Note that the true value of  $e^{0.5} = 1.648721$ . Add terms until the absolute value of the approximate error estimate  $\varepsilon_a$  falls below a pre-specified error criterion  $\varepsilon_s$  conforming to three significant digits.



# Example 3.2

- Solution: The error criterion that ensures a result is correct to at least 3 significant digits:

$$\varepsilon_s = (0.5 \times 10^{2-3})\% = 0.05\%$$

First estimation is simply 1, then adding second term  $e^x = 1 + x$

For  $x = 0.5$ ;  $e^{0.5} = 1 + 0.5 = 1.5$

True percent relative error:

$$\varepsilon_t = \frac{1.648721 - 1.5}{1.648721} 100\% = 9.02\%$$

Approximate estimate of the error:

$$\varepsilon_a = \frac{1.5 - 1}{1.5} 100\% = 33.3\%$$

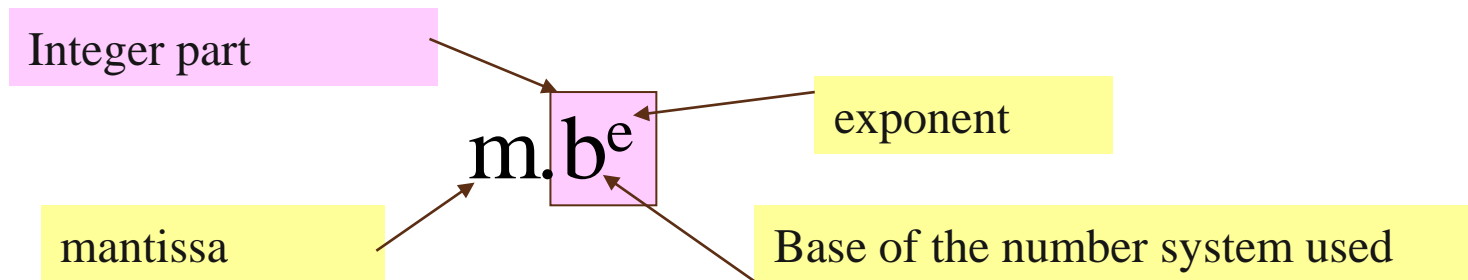
Continue until  $\varepsilon_a < \varepsilon_s$

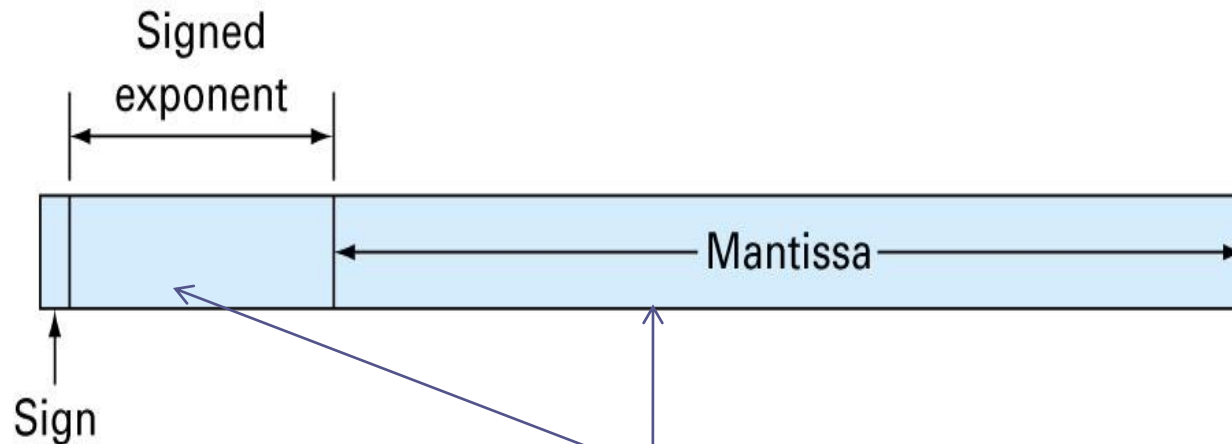
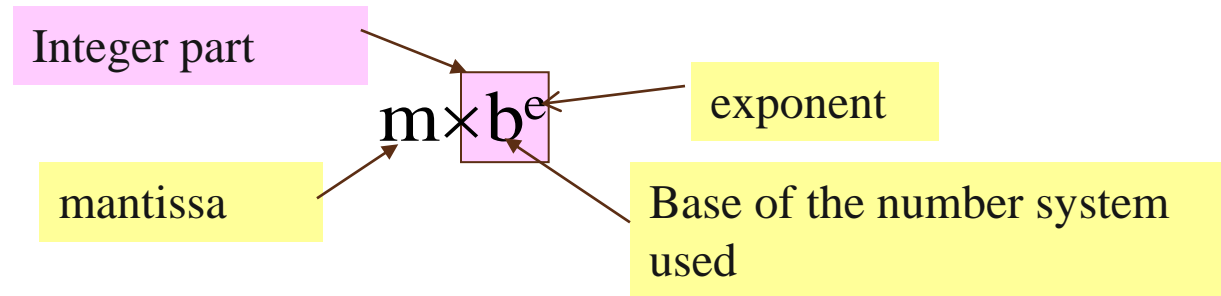
Terms	Result	$\varepsilon_a$ (%)
1	1	
2	1.5	33.3
3	1.625	7.69
4	1.645833333	1.27
5	1.648437500	0.158
6	1.648697917	0.0158



# Round-off Errors

- Numbers such as  $\pi$ ,  $e$ , or  $\sqrt{7}$  cannot be expressed by a fixed number of significant figures.
- Computers use a base-2 representation, they cannot precisely represent certain exact base-10 numbers.
- Fractional quantities are typically represented in computer using “floating point” form, e.g.,





$$156.78 \Leftrightarrow 0.15678 \times 10^3$$

$$\frac{1}{34} = 0.029411765$$

$$0.0294 \times 10^0$$

Suppose only 4  
decimal places to be stored

- Normalized to remove the leading zeroes. Multiply the mantissa by 10 and lower the exponent by 1

$$0.294\underline{1} \times 10^{-1}$$

Additional significant figure is retained



$$\frac{1}{b} \leq |m| < 1$$

Therefore

for a base-10 system  $0.1 \leq m < 1$

for a base-2 system  $0.5 \leq m < 1$

- Floating point representation allows both fractions and very large numbers to be expressed on the computer. However,
  - Floating point numbers take up more room.
  - Take longer to process than integer numbers.
  - Round-off errors are introduced because mantissa holds only a finite number of significant figures.



# Chopping

Example:

$\pi=3.14159265358$  to be stored on a base-10 system carrying 7 significant digits.

$\pi=3.141592$       chopping error       $\varepsilon_t=0.00000065$

## If rounded

$$\pi=3.141593 \qquad \varepsilon_t=0.00000035$$

- Some machines use chopping, because rounding adds to the computational overhead. Since number of significant figures is large enough, resulting chopping error is negligible.

