Universiti Malaysia PAHANG
Engineering • Technology • Creativity

# NUMERICAL METHODS & OPTIMISATION

## Part II: Modelling & Error Analysis

Raihana Edros
Faculty of Engineering Technology
rzahirah@ump.edu.my

# Chapter Description

- Aims
  - Apply numerical methods in solving engineering problem and optimisation

- Expected Outcomes
  - Apply both exact and numerical solutions in mathematical modelling to solve engineering problem.
  - Calculate true percent error, percent relative error & percent tolerance
  - Estimate the truncation error by using Taylor Series

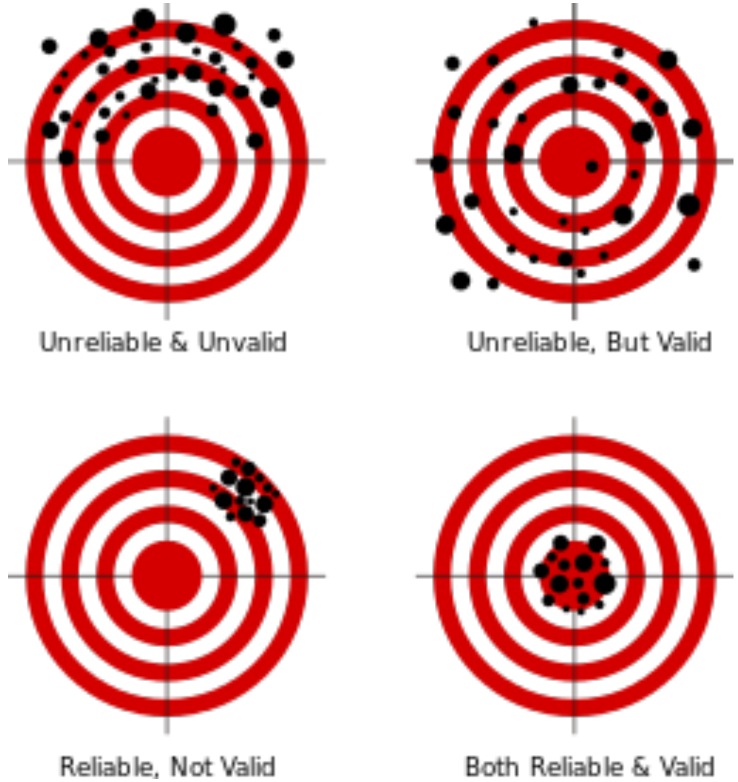
- References
  - Steven C. Chapra and Raymond P. Canale (2009), Numerical Methods for Engineers, McGraw-Hill, 6th Edition

# Approximation & round-off errors

- The numerical methods usually yield estimates that are closed to the exact analytical solution
- Though the values are closed, but they are not the same
- The discrepancy between the two values is called error
- Numerical methods involved approximation to the exact values using mathematical model
- Errors give information on how accurate is the numerical methods used to solve engineering problems

# Accuracy & precision

- Errors can be characterized based on their accuracy & precision
- Precision: How closely the individual computed or measured values agree with each other
- Accuracy: How closely a computed or measured value agrees to the true value



# Error definitions

- Error definitions:
  - Truncation errors: when approximations are used to represent exact mathematical procedures (e.g.: Taylor series)
  - Round-off errors: when numbers having limited significant errors are used to represent exact numbers
  - For both types:

$$True = approximation + error \qquad E_t = true - approximation$$

  - $E_t$ is the true error

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

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

# Error definitions (cont'd)

- In most of engineering problems, the true value remains unknown – error has to be estimated in the absence of the true value
- Certain numerical methods use iterative approaches to compute answer where a present approximation is made on the basis of a previous approximation
- The approximation error is given by:

$$\varepsilon_a = \frac{current\ approximation - previous\ approximation}{current\ approximation} \times 100\%$$

- $\varepsilon_a$ can be either positive or negative value; rather, an absolute value $|\varepsilon_a|$

# Error definitions (cont'd)

- The iteration will continue until:

$$|\varepsilon_a| < \varepsilon_S$$

where $\varepsilon_S$ is the prespecified acceptable percent tolerance and can be determined using:

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

where $n$ is the significant figures

# Round-off errors

- Originate from the fact that computers retain only a fixed number of significant figures
- Therefore, the numbers can not be represented exactly by the computer
- The discrepancy introduced by omission of significant figures is called round-off errors
- Computer representation of numbers:
  - Integer representation
  - Floating-point representation

# Round-off errors (cont'd)

- Number systems:
  - A convention for representing quantities
  - The most popular – decimal or base-10 number system
  - 'base' – the number used as the reference for constructing the system & uses the 10 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 to represent numbers
  - Computer has base-2 number system i.e. the binary code, 1 and 0
  - E.g.: binary system = 11 is equivalent to $(1 \times 2^1) + (1 \times 2^0) = 2+1 = 3$

# Round-off errors – Integer representation

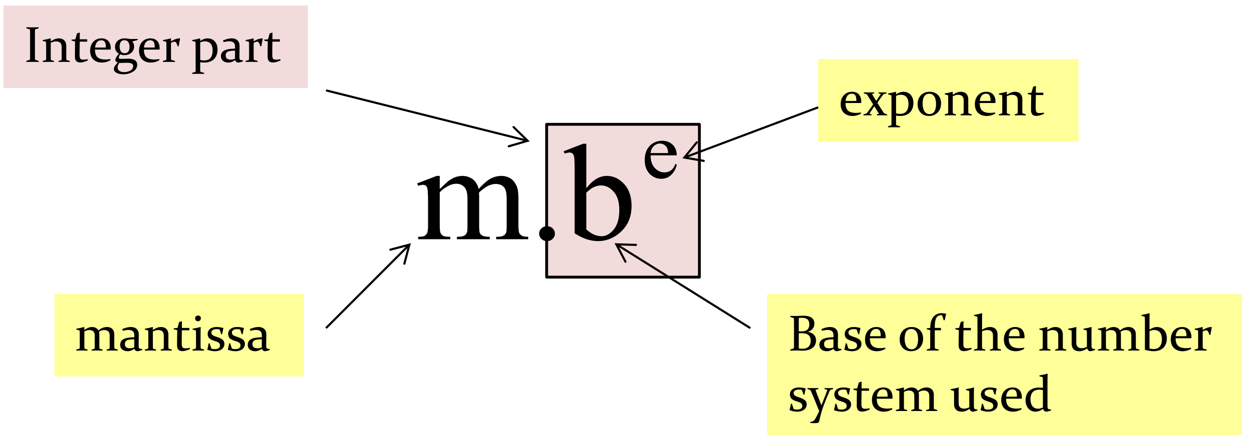
- Base-10 numbers can be represented by base-2 system: how numbers are represented on a computer
- Signed magnitude representation: first bit (leftmost) indicates the sign (i.e. 1 for positive & 0 for negative) while the rest indicates the magnitude
- E.g.:

  1001 1000 → total 8 bits: first bit 1 → positive while $2^5 + 2^4 = 32+8 = 40$

  Thus: The magnitude is 40 in 7-bit binary
- 2's complement technique: incorporate the sign into the number's magnitude
- Computers have limited capability to represent integers: numbers out of range can't be represented, so does the fractional quantities – floating-point

# Round-off errors – Floating-point representation

- Fractional quantities are represented in computers using floating-point form
- The number is expressed as a fractional part – mantissa or significand - and fractional part – exponent or characteristic:



# Round-off errors – Floating-point representation

- Significant figures are those number can be used with confidence.
- E.g: 0.0123
  0.00123
  0.000123
- All above are three significant figures
- Zeros are not always significant figures because they may necessary just to locate a decimal point.
- E.g: 45700
- 3,4 or 5 significant figures depend whether zeros are known with confidence.

# Round-off errors – Floating-point representation

- Using scientific notation:

  $4.57 \times 10^4$ - 3 significant figures

  $4.570 \times 10^4$ - 4 significant figures

  $4.5700 \times 10^4$ - 5 significant figures

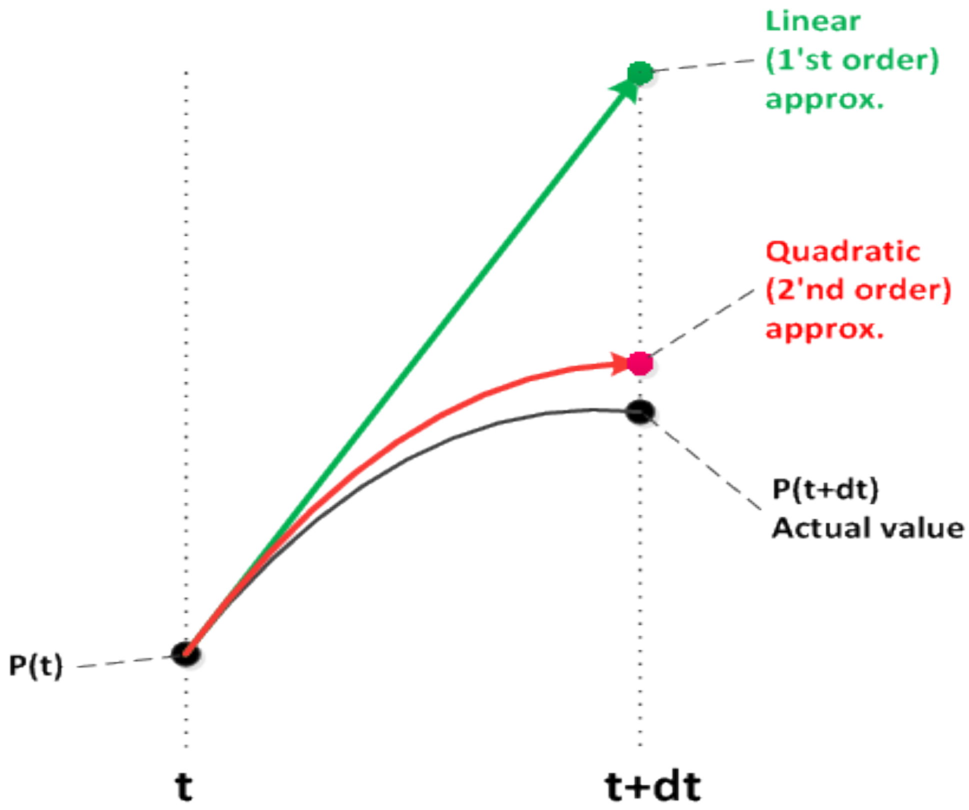
- Generally:
  - Non zero digit are always significant e.g : 486912 (6SF)
  - Any zero between two significant numbers are significant e.g :708 (3SF)
  - Final zero or trailing zero in a decimal portion only significant e.g 0.008980 (4SF)

# Truncation errors – The Taylor Series

- Those that results from using an approximation in place of an exact mathematical procedure – the difference equation only approximates the true value of the derivative
- Mathematical formulation that is used widely to express functions in an approximate fashion
- It was built term by term, started with zero-order approximation.
- The higher the order of approximation applied, the lower the truncation error

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 \\ &\quad + \frac{f'''(x_0)}{3!}(x-x_0)^3 + \frac{f''''(x_0)}{4!}(x-x_0)^4 + \cdots \\ &= \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n. \end{aligned}$$

# Truncation errors – The Taylor Series




Source: https://www.clear.rice.edu/comp130/12spring/predprey/

# Truncation errors – The Taylor Series (cont'd)

- The n-th order of Taylor series expansion will be exact for the n-th order polynomial but not for:
  - Differentiable functions
  - Continuous functions
- Each additional terms will contribute to some improvement to the approximation
- The error is proportional to the step size, h but not the number of terms added - lower number of terms gives better approximation – (Example 4.2)

# The remainder of Taylor Series Expansion

- The lower order derivatives account for a greater share of the remainder than the higher order terms
- It is inconvenient to deal with the remainder in infinite series format, one simplification can be used to truncate the remainder itself to:

$$R_1 = \frac{f''(\xi)}{2!} h^2$$

# Numerical Differentiation

- In order to solve the derivatives of an equation, numerical differentiation of Taylor Series can be used with the following approximations:
  - Forward difference approximation

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} + O(x_{i+1} - x_i)$$

  - Backward difference approximation

$$f'(x_i) \cong \frac{f(x_i) - f(x_{i-1})}{h}$$

  - Centered difference approximation

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - O(h^2)$$

# Conclusion

- Percent error, percent relative error & percent tolerance can be calculated for engineering problems
- Truncation error can be estimated by using Taylor Series

## Main Reference

Steven C. Chapra and Raymond P. Canale (2009), Numerical Methods for Engineers, McGraw-Hill, 6th Edition

Any enquiries kindly contact:
Raihana Edros, PhD
rzahirah@ump.edu.my