Round-Off and Truncation Errors
Accuracy and Precision

• Accuracy: how closely a computed or measured value agrees with the true values.

• Precision: how closely individual computed or measured values agree with each other.
Accuracy and Precision (cont.)

- An example from marksmanship illustrating the concepts of accuracy and precision: (a) inaccurate and imprecise, (b) accurate and imprecise, (c) inaccurate and precise, and (d) accurate and precise.
Error Definitions

• True error: $E_t = \text{true value} - \text{approximation}$.
• True fractional relative error: $\varepsilon_t = \frac{\text{true value} - \text{approximation}}{\text{true value}} \times 100\%$.
• Approximate fractional relative error: $\varepsilon_a = \frac{\text{approximate error}}{\text{approximation}} \times 100\%$.
• Or, $\varepsilon_a = \frac{\text{present approx.} - \text{previous approx.}}{\text{approximation}} \times 100\%$. 
Round-off Errors

- Computers cannot represent some quantities exactly.
- The size and precision of numbers are limited in a computer.
- Lead to erroneous results.
- Certain numerical manipulations are highly sensitive to round-off errors.
Round-off Errors (cont.)

• Round-off errors will accumulate in a series of computation. Example:

```matlab
s=0;
for i=1:10000
    s=s+0.0001;
end
```
Examples

- Irrational numbers: pi, sqrt(2), and etc. That is, infinite precision. Cannot be solved.
- Rational numbers: 1/3, 0.1, and etc. That is, numbers that cannot be expanded by binary digits. Cannot by solved.
- Range exceeds. Too large or to small. Can be solved by increasing the size.
- Precision exceeds. Too many digits. Can be solved by increasing the size.
IEEE 754 Floating Number Range

- Single precision: 32-bit, $10^{-38}$ to $10^{39}$. 7 decimal digit precision.
- Double precision: 64-bit, $10^{-308}$ to $10^{308}$. 15 decimal digit precision.
IEEE 751 Single-Precision Floating-Point Data Types

- Format: 32 bits
- S bit: sign bit. 1 for positive, 0 for negative.
- Fraction: 23 bits.
- Normalized: Value=\((-1)^S \times 1.fraction \times 2^{(exp-127)}\)
IEEE 751 Single-Precision Floating-Point Data Types (cont.)

• Example

$$1995 = 1.1111001011 \times 2^{1010}$$
IEEE 751 Double-Precision Floating-Point Data Types

- Format: 64 bits
- Fraction: 52 bits.
Round-off Errors due to Arithmetic Operations

• Addition or subtraction of a large and small number. Example:
  \[0.1234567 + 0.0000000001234567 = 0.1234567\]

• Multiplication or division. Example:
  \[0.1234567 \times 0.1234567 = 0.01524156 \text{ (exact 0.01524155677489)}\]
Truncation Errors

- Result from using an approximation in place of an exact mathematical procedure. Example:

\[
\frac{dv}{dt} \approx \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}
\]
Taylor Series

• Any function $f(x)$ can be infinitely approximated near a point $x_0$ by a polynomial as follow

$$f(x) = f(x_0) + f(x_0)'(x-x_0) + \frac{1}{2!}f(x_0)''(x-x_0)^2 + \frac{1}{3!}f(x_0)'''(x-x_0)^3 + \cdots$$

• All derivatives of the Taylor series are the same as the original function at $x_0$. 
Taylor Series: Example

The approximation of \( f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2 \) at \( x = 1 \) by zero-order, first-order, and second-order Taylor series expansions.
Taylor Series: Example (cont.)

- $R_n$ is the remainder terms
- $R_n = O(h^{n+1})$

\[
f(x_{i+1}) = f(x_i) + f(x_i)'h + \frac{1}{2!}f(x_i)''h^2 + \cdots + \frac{1}{n!}f(x_i)^{(n)}h^n + R_n
\]
Example 4.3

• Let \( f(x) = \cos x \), \( x_i = \pi/4 \) and \( x_{i+1} = \pi/3 \). Approximate \( f(x_{i+1}) \) by Taylor series.

| Order \( n \) | \( f^{(n)}(x) \) | \( f(\pi/3) \) | \( |e_i| \) |
|--------------|-----------------|-----------------|--------|
| 0            | \( \cos x \)    | 0.707106781     | 41.4   |
| 1            | \(-\sin x\)     | 0.521986659     | 4.40   |
| 2            | \(-\cos x\)     | 0.497754491     | 0.449  |
| 3            | \( \sin x \)    | 0.499869147     | \( 2.62 \times 10^{-2} \) |
| 4            | \( \cos x \)    | 0.500007551     | \( 1.51 \times 10^{-3} \) |
| 5            | \(-\sin x\)     | 0.500000304     | \( 6.08 \times 10^{-5} \) |
| 6            | \(-\cos x\)     | 0.499999988     | \( 2.44 \times 10^{-6} \) |
Error of First Derivative by Finite Difference

• The error is $O(x_{i+1}-x_i)$

$$f(x_{i+1}) = f(x_i) + f(x_i)'(x_{i+1} - x_i) + R_1$$

$$f(x_i)' = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} - \frac{R_1}{x_{i+1} - x_i} = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} + O(x_{i+1} - x_i)$$
Numerical Differentiation by Finite Difference

• Forward:
  \[ f(x_i)' = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h) \]

• Backward:
  \[ f(x_i)' = \frac{f(x_i) - f(x_{i-1})}{h} + O(h) \]

• Mid value:
  \[ f(x_i)' = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + O(h^2) \]
Example 4.4

- Estimate the error of the first derivatives at \( x=0.5 \), \( h=0.5 \) and \( h=0.25 \).

\[ f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2 \]

- \( h=0.5 \)
  - Forward: 58.9\%, Backward: 39.7\%, Mid value: 9.6\%

- \( h=0.25 \)
  - Forward: 26.5\%, Backward: 21.7\%, Mid value: 2.4\%
Numerical Differentiation by Finite Difference (cont.)

- Graphical depiction of (a) forward, (b) backward, and (c) centered finite-divided-difference approximations of the first derivative.
Total Numerical Errors

- Truncation error: the smaller step size, the smaller.
- Round-off error: the smaller step size, the more arithmetic operations, the larger.
- Trade off.
- Centered difference approximation:
  - Truncation error: $O(h^2)$
  - Roundoff error: $O(1/h)$
Total Numerical Errors (cont.)
Example 4.5

• Estimate the error of the first derivatives at $x=0.5$, from $h=1$ to $h=10^{-10}$ by centered difference approximation.

$$f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$$
Example 4.5 (cont.)

Plot of error versus step size

[Graph showing the relationship between error and step size, with error on a logarithmic scale and step size also on a logarithmic scale.]