Lecture # 02

TYPES & MEASUREMENT OF ERRORS

REVIEW OF LAST CLASS:

Under Error Measurement, We discussed,

i. **Absolute(True) Error**

ii. **Relative & Percentage Relative Error**

Note:

The relative error is generally a better measure of accuracy than the absolute error because it takes into consideration the size of the number being approximated.

Example:

For example, suppose that you have the task of measuring the lengths of a bridge and a rivet and come up with 9999 and 9 cm, respectively. If the true values are 10,000 and 10 cm, respectively, the error in both cases is 1 cm. However, their percent relative errors can be computed using Eq. (4.3) as 0.01% and 10%, respectively. Thus, although both measurements have an absolute error of 1 cm, the relative error for the rivet is much greater. We would probably conclude that we have done an adequate job of measuring the bridge, whereas our estimate for the rivet leaves something to be desired.

Approximate Errors :

$$\varepsilon_a = \frac{\text{present approximation} - \text{previous approximation}}{\text{present approximation}} 100\%$$

For iterative approximations, continue to iterate until the relative approximate error magnitude is less than a specified ***stopping criterion***

$$\boxed{|\varepsilon_a| < \varepsilon_s}$$

For accuracy to ***at least n significant figures*** set the stopping criterion to

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

Types of Errors :

Errors are of two types:

1. Truncation Error
2. Round-off Error

1. **Truncation error** is a result of using approximations to represent exact mathematical procedures ***OR*** *when an iterative method is terminated*

2. **Round-off error** occurs when only certain digits and decimal places are used to represent exact numbers.

Truncation Error:

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

HINT:

9.8.1 DEFINITION If f has derivatives of all orders at x_0, then we call the series

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x-x_0)^k = f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \cdots + \frac{f^{(k)}(x_0)}{k!}(x-x_0)^k + \cdots \quad (1)$$

the ***Taylor series for f about $x = x_0$***. In the special case where $x_0 = 0$, this series becomes

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!}x^k = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \cdots + \frac{f^{(k)}(0)}{k!}x^k + \cdots \quad (2)$$

in which case we call it the ***Maclaurin series for f***.

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

Starting with the simplest version, $e^x = 1$, add terms one at a time in order to estimate $e^{0.5}$. After each new term is added, compute the true and approximate percent relative errors

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

Starting with the simplest version, $e^x = 1$, add terms one at a time in order to estimate $e^{0.5}$. After each new term is added, compute the true and approximate percent relative errors

Solution. First to determine the error criterion that ensures a result that is correct to at least three significant figures:

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

Thus, we will add terms to the series until ε_a falls below this level.

$$e^x = 1 + x$$

or for $x = 0.5$

$$e^{0.5} = 1 + 0.5 = 1.5$$

This represents a true percent relative error

$$\varepsilon_t = \left|\frac{1.648721 - 1.5}{1.648721}\right| \times 100\% = 9.02\%$$

to determine an approximate estimate of the error, as in

$$\varepsilon_a = \left|\frac{1.5 - 1}{1.5}\right| \times 100\% = 33.3\%$$

Because ε_a is not less than the required value of ε_s, we would continue the computation by adding another term, $x^2/2!$, and repeating the error calculations. The process is continued until $|\varepsilon_a| < \varepsilon_s$. The entire computation can be summarized as

Terms	Result	ε_t, %	ε_a, %
1	1	39.3	
2	1.5	9.02	33.3
3	1.625	1.44	7.69
4	1.645833333	0.175	1.27
5	1.648437500	0.0172	0.158
6	1.648697917	0.00142	0.0158

Representation of Real Numbers:

1. Binary Machine Numbers:

A 64-bit (binary digit) representation is used for a real number (according to IEEE standards).

$$(-1)^s 2^{c-1023}(1+f)$$

This representation is called floating point representation.

The first bit is a sign indicator, denoted s. This is followed by an 11-bit exponent, c, called the **characteristic**, and a 52-bit binary fraction, f, called the **mantissa**. The base for the exponent is 2.

Examples:

0 10000000011 1011100100010000000000000000000000000000000000000000

Sign | Characteristic | Mantissa

Sign: (0: positive; 1:negative)

$$c = 1 \cdot 2^{10} + 0 \cdot 2^9 + \cdots + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1024 + 2 + 1 = 1027.$$

The exponential part of the number is, therefore:

$$2^{1027-1023} = 2^4$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Mantissa The final 52 bits is:

$$f = 1 \times \left(\frac{1}{2}\right)^1 + 1 \times \left(\frac{1}{2}\right)^3 + 1 \times \left(\frac{1}{2}\right)^4 + 1 \times \left(\frac{1}{2}\right)^5 + 1 \times \left(\frac{1}{2}\right)^8 + 1 \times \left(\frac{1}{2}\right)^{12}$$

As a consequence, this machine number precisely represents the decimal number

$$(-1)^s 2^{c-1023}(1+f) = (-1)^0 \cdot 2^{1027-1023}\left(1 + \left(\frac{1}{2} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \frac{1}{4096}\right)\right)$$

$$= 27.56640625.$$

Reading Activity: **Page # 18 & 19** of Textbook

Overflow & Underflow

An overflow error is produced when trying to use a number too large (greater than the corresponding R_{max}):

- In most computers, execution is aborted.
- IEEE format may support them by assigning the symbolic values

$$\pm\infty \quad \text{or} \quad NaN.$$

An underflow error is produced when trying to use a number too small (less, in absolute value, than the corresponding R_{min}). Two possible behaviors:

- It lies in the range of denormalized numbers, so it is still representable. In this case, precision decreases and it is called gradual underflow.
- Otherwise, it is identified to 0.

2. Decimal Machine Numbers: (Normalized Floating Point Representation)

$\boxed{\pm 0.d_1d_2 \ldots d_k \times 10^n}$, $\quad 1 \le d_1 \le 9, \quad 0 \le d_i \le 9$

Any positive real number within the numerical range of the machine can be normalized to the form:

$$y = 0.d_1d_2 \ldots d_kd_{k+1}d_{k+2} \ldots \times 10^n$$

The floating-point form of y, denoted $fl(y)$, is obtained by terminating the mantissa of y at k decimal digits. This can be performed by using one of two methods:

1. **Chopping:**

$$fl(y) = 0.d_1 d_2 \ldots d_k \times 10^n$$

2. **Rounding:**

$$fl(y) = 0.\delta_1 \delta_2 \ldots \delta_k \times 10^n$$

Example: Convert the following numbers to 4-digit by chopping and rounding:

$$x = 635894 \; , \; y = 0.00218 \; , \; z = 584.63$$

Chopping:

$$x^* = 0.6358 \times 10^6$$

Rounding:

$$x^* = 0.6359 \times 10^6$$

Similarly do for y & z

Chopping: $y^* = 0.2180 \times 10^{-2}, \quad z^* = 0.5486 \times 10^{3}$

Rounding: $y^* = 0.2180 \times 10^{-2}, \quad z^* = 0.5486 \times 10^{3}$

Definition 1: Suppose that p^* is an approximation to p. The **absolute error** is $e_p = |p - p^*|$, and the **relative error** is $\delta_p = \frac{|p-p^*|}{|p|}$ provided that $p \neq 0$.

Determine the absolute and relative errors when approximating p by p^* when

(a) $p = 0.3000 \times 10^1$ and $p^* = 0.3100 \times 10^1$;

(b) $p = 0.3000 \times 10^{-3}$ and $p^* = 0.3100 \times 10^{-3}$;

(c) $p = 0.3000 \times 10^4$ and $p^* = 0.3100 \times 10^4$.

Solution

(a) For $p = 0.3000 \times 10^1$ and $p^* = 0.3100 \times 10^1$ the absolute error is 0.1, and the relative error is $0.333\overline{3} \times 10^{-1}$.

(b) $p = 0.3000 \times 10^{-3}$ and $p^* = 0.3100 \times 10^{-3}$;

(c) $p = 0.3000 \times 10^{4}$ and $p^* = 0.3100 \times 10^{4}$.

(b) For $p = 0.3000 \times 10^{-3}$ and $p^* = 0.3100 \times 10^{-3}$ the absolute error is 0.1×10^{-4}, and the relative error is $0.333\overline{3} \times 10^{-1}$.

(c) For $p = 0.3000 \times 10^{4}$ and $p^* = 0.3100 \times 10^{4}$, the absolute error is 0.1×10^{3}, and the relative error is again $0.333\overline{3} \times 10^{-1}$.

Finite Digit Arithmetic:

Example 3 Suppose that $x = \frac{5}{7}$ and $y = \frac{1}{3}$. Use five-digit chopping for calculating $x + y$, $x - y$, $x \times y$, and $x \div y$.

$$x = \frac{5}{7} = 0.\overline{714285} \quad \text{and} \quad y = \frac{1}{3} = 0.\overline{3}$$

$$x \oplus y = fl(fl(x) + fl(y)) = fl\left(0.71428 \times 10^0 + 0.33333 \times 10^0\right)$$
$$= fl\left(1.04761 \times 10^0\right) = 0.10476 \times 10^1.$$

Error Analysis:

The true value is $x + y = \frac{5}{7} + \frac{1}{3} = \frac{22}{21}$, so we have

$$\text{Absolute Error} = \left| \frac{22}{21} - 0.10476 \times 10^1 \right| = 0.190 \times 10^{-4}$$

and

$$\text{Relative Error} = \left| \frac{0.190 \times 10^{-4}}{22/21} \right| = 0.182 \times 10^{-4}.$$

Operation	Result	Actual value	Absolute error	Relative error
$x \oplus y$	0.10476×10^1	22/21	0.190×10^{-4}	0.182×10^{-4}
$x \ominus y$	0.38095×10^0	8/21	0.238×10^{-5}	0.625×10^{-5}
$x \otimes y$	0.23809×10^0	5/21	0.524×10^{-5}	0.220×10^{-4}
$x \oplus y$	0.21428×10^1	15/7	0.571×10^{-4}	0.267×10^{-4}