Approximation and Errors in numerical computations:

(1) **Approximate numbers:** There are two types of numbers exact and approximate. Exact numbers are 2, 4, 9, 13, 7/2, 6.45, etc. But there are numbers such as 4/3 (= 1.3333...), $\sqrt{2}$ (= 1.414213...) and π (= 3.141592...) which cannot be expressed by a finite number of digits. These may be approximated by numbers 1.3333, 1.4142 and 3.1416 respectively. Such numbers which represent the given numbers to a certain degree of accuracy are called approximate numbers.

(2) **Significant Digits of Precision:** Significant digits are digits beginning with the leftmost nonzero digit and ending with the rightmost correct digit, including final zeros that are exact.

Ex: 7845, 3.589, 0.4758 contains four significant digits

Ex: 0.00386, 0.000587, 0.00203 contains only three significant digits.

(3) **Accuracy and Precision:** Accurate to n decimal places means that you can trust n digits to the right of the decimal place. Accurate to n significant digits means that you can trust a total of n digits as being meaningful beginning with the leftmost nonzero digit.

(4) **Rounding off:** There are numbers with large number of digits eg., 22/7 = 3.14285714 3. In practice, it is desirable to limit such numbers to a manageable number of digits such as 3.14 or 3.143. This process of dropping unwanted digits is called rounding off.

Rule to round off a number

A number is rounded to position n by the following rule:

(i) Discard all digits to the right of the nth digit

(ii) If this discarded number is

(a) less than half a unit in the nth place, leave the nth digit unchanged;

(b) greater than half a unit in the nth place, increase the nth digit by unity;

(c) exactly half a unit in the nth place, increase the nth digit by unity if it is odd otherwise leave it unchanged.

For example (i) 84767 rounded to three significant figures = 84800

(ii) 3.567 rounded to three significant figures = 3.57

(iii) 8.73500 rounded to two decimal places = 8.74

(iv) 7.24500 rounded to two decimal places = 7.24

(v) 11.34576523 rounded to five decimal places = 11.34577

Errors: In any numerical computation we come across the following types of errors:

(1) **Inherent errors:** Errors which are already present in the statement of a problem before its solution are called inherent errors. Such errors arise either due to the given data being approximate or due to the limitation of mathematical tables, calculators or the digital computer. Inherent errors can be minimized by taking better data or by using high precision computer aids.

(2) Rounding errors: These errors arise from the process of rounding off the numbers during the computation. Such errors are unavoidable in most of the calculations due to the limitations of the computing aids. Rounding errors can, however be reduced:

(i) by changing the calculation procedure so as to avoid subtraction of nearly equal numbers or division by a small number;

(ii) by retaining at least one more significant figure at each step than that given in the data and rounding off at the last step.

(3) Truncation errors: These errors are caused by using approximate results or on replacing an infinite process by a finite one. For example, we consider the Taylor series expansion of $f(x)$ about $x = c$, $c \in [a,b]$. If we retain the first n terms, we get the approximation

$$f(x) \approx f(c) + (x-c)f'(c) + \frac{(x-c)^2}{2!}f''(c) + \cdots + \frac{(x-c)^{n-1}}{(n-1)!}f^{(n-1)}(c) \quad \text{---(1)}$$

and the truncation error (T.E.) is given by

$$T.E. = \frac{(x-c)^n}{n!}f^{(n)}(\xi), \text{ where } \xi \text{ lies between } c \text{ and } x$$

$$\Rightarrow |T.E.| \leq \frac{1}{n!} M_n \cdot \max_{[a,b]} |x-c|^n \quad \text{where } M_n = \max_{[a,b]} |f^{(n)}(x)| \quad \text{---(2)}$$

Assume that the value of M_n or its estimate is available. Then we can use it to determine an upper bound on the error.

Note: Suppose that we require $|T.E.| \leq \varepsilon$. Then we can determine (i) the number of terms (n) for a given x and ε by eq. (2).

(ii) $|x-c|$ for a given n and ε. This gives an interval about c in which the Taylor polynomial approximation given by eq (1) is valid to the prescribed accuracy.

(4) Absolute, Relative and Percentage errors: If X is the true value of a quantity and X^* is its approximate value, then

$$\text{Error} = \text{True value} - \text{Approximate value} = X - X^*$$

$$\text{Absolute Error} = |\text{True value} - \text{Approximate value}| = |X - X^*|$$

$$\text{Relative Error} = \frac{|X - X^*|}{|X|}$$

(True value is also known as exact value)

$$\text{Percentage error} = \frac{|X - X^*|}{|X|} \times 100$$

Note: (1) The relative and percentage errors are independent of the units used while absolute error is expressed in terms of these units.

(2) If a number is correct to n decimal places then the error

$$= \frac{1}{2} \times 10^{-n}$$

For Ex: If the number is 3.1416 correct to 4 decimal places, then the error $= \frac{1}{2} \times 10^{-4} = 0.00005$

(3) For practical reasons, the relative error is usually more meaningful than the absolute error. For example

if $x_1 = 1.333$, $x_1^* = 1.334$, and $x_2 = 0.001$, $x_2^* = 0.002$, then the absolute error of x_1^* as an approximation to x_1

$$= |x_1 - x_1^*| = |1.333 - 1.334| = 0.001 = 10^{-3}$$

and the absolute error of x_2^* as an approximation to x_2

$$= |x_2 - x_2^*| = |0.001 - 0.002| = 0.001 = 10^{-3}$$

Both are same.

But the relative error of $x_1^* = \frac{|x_1 - x_1^*|}{|x_1|} = \frac{1}{1.333} \times 10^{-3}$

$$= \frac{0.001}{1.333} = 0.750 \times 10^{-3}$$

and the relative error of x_2^*

$$= \frac{|x_2 - x_2^*|}{|x_2|} = \frac{0.001}{0.001} = 1$$

which indicates that x_1^* is a good approximation to x_1 but x_2^* is a poor approximation to x_2.

Note: If the approximate value of a number X having n decimal digits is X^* then

(i) Relative error due to rounding off to k digits

$$= \frac{|X - X^*|}{|X|} < \frac{1}{2} \cdot 10^{1-k}$$

(ii) Relative error due to truncation to k digits

$$= \frac{|X - X^*|}{|X|} < 10^{1-k}$$

Que Round off the numbers 865250 and 37.46235 to four significant figures and compute absolute error, relative error, percentage error in each case.

Sol (i) Number rounded off to four significant figures

$= 865200$

Absolute error = | True value − Approximate value |

$= |865250 - 865200| = 50$

$$\text{Relative Error} = \frac{|\text{True value} - \text{Appr. value}|}{|\text{True value}|} = \frac{50}{865250} = 6.71 \times 10^{-5}$$

Percentage Error = Relative Error $\times 100 = 6.71 \times 10^{-3}$

(ii) Number rounded off to four significant figures = 37.46

$\therefore$ Absolute Error = $|37.46235 - 37.46| = 0.00235$

$$\text{Relative Error} = \frac{0.00235}{37.46235} = 6.27 \times 10^{-5}$$

Percentage Error = $6.27 \times 10^{-5} \times 100 = 6.27 \times 10^{-3}$

Que Find the absolute error if the number $X = 0.00545828$ is

(i) truncated to three decimal digits

(ii) rounded off to three decimal digits

Sol Given $X = 0.00545828 = 0.545828 \times 10^{-2}$

(i) After truncated to three decimal digits, its approximate value $X^* = 0.545 \times 10^{-2}$

$\therefore$ Absolute error = $|X - X^*| = 0.000828 \times 10^{-2}$

$= 0.828 \times 10^{-5}$ A

(ii) Given $X = 0.0054582 8$

$= 0.545828 \times 10^{-2}$

After rounded off to three decimal places, its approximate value $X^* = 0.546 \times 10^{-2}$

$\therefore$ Absolute Error $= |X - X^*|$

$= |0.545828 \times 10^{-2} - 0.546000 \times 10^{-2}|$

$= 0.000172 \times 10^{-2}$

$= 0.172 \times 10^{-5}$ Ans

Que Find the relative error if the number $X = 0.004997$ is

(i) truncated to three decimal places

(ii) rounded off to three decimal digits

Sol We have $X = 0.004997 = 0.4997 \times 10^{-2}$

(i) After truncated to three decimal places, its approximate value $X^* = 0.499 \times 10^{-2}$

$\therefore$ Relative error $= \dfrac{|X - X^*|}{|X|}$

$= \dfrac{|0.4997 \times 10^{-7} - 0.499 \times 10^{-2}|}{|0.4997 \times 10^{-2}|}$

$= \dfrac{0.0007}{0.4997} = 0.00140 = 0.140 \times 10^{-2}$ Ans

(ii) After rounded off to three decimal places, its approximate value $X^* = 0.005 = 0.500 \times 10^{-2}$

$\therefore$ Relative error $= \dfrac{|X - X^*|}{|X|}$

$$= \frac{|0.4997 \times 10^{-2} - 0.5000 \times 10^{-2}|}{|0.4997 \times 10^{-2}|}$$

$$= \frac{0.0003}{0.4997} = 0.000600 = 0.600 \times 10^{-3} \quad \text{Ans}$$

Que Using Taylor series expansion of e^{-x} about $c = 0$. Determine

(i) maximum error for $x \in [-1, 1]$, when the first four terms are used in the approximation. Also find the maximum error when $x = 0.3$.

(ii) the least number of terms required in the approximation such that $|\text{error}| \leq 5 \times 10^{-4}$ for $x \in [-1, 1]$.

(iii) x, when the approximation obtained from the first four terms is accurate to 5×10^{-4}.

Sol Let $f(x) = e^{-x}$. Then $f(0) = 1$

$f^{(n)}(x) = (-1)^n e^{-x}$ $f^{(n)}(0) = (-1)^n$ and $f^{(n)}(\xi) = (-1)^n e^{-\xi}$

$\therefore$ Taylor series expansion of e^{-x} about $c = 0$ is given by

$$f(x) = f(0) + x f'(0) + \frac{x^2}{2!} f''(0) + \cdots + \frac{x^{n-1}}{(n-1)!} f^{n-1}(0) + E_n$$

where $E_n = \dfrac{x^n}{n!} f^{(n)}(\xi)$ where ξ lies between 0 and x —(1)

$\therefore$ When the first four terms are used, Taylor series expansion of e^{-x} is

$$e^{-x} \sim 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!}$$

and Error $= E_4 = \frac{x^4}{4!} f^{(4)}(\xi)$ where ξ lies between 0 and x

$\Rightarrow |\text{Error}| = \left| \frac{x^4}{4!} e^{-\xi} \right|$

$\leq \frac{e}{24}$ ($\because |x| \leq 1$ and $e^{-\xi} \leq e$ as $x \in [-1, 1]$ and ξ lies between 0 and x)

$\Rightarrow |\text{Error}| \leq 0.1133$

$\therefore$ Maximum error $= 0.1133$ A

For $x = 0.3$, $|\text{Error}| = \left| \frac{(0.3)^4}{4!} e^{-\xi} \right| \leq \frac{(0.3)^4}{24} e$ ($\because e^{-\xi} \leq e$ as above)

$= 0.00092$

$\therefore$ Maximum error $= 0.00092$ when $x = 0.3$ A

(ii) By eqn. (1), if n terms are required in the approximation, then Error $= \frac{x^n}{n!} f^{(n)}(\xi)$ where ξ lies between 0 and x

$\Rightarrow |\text{Error}| = \left| \frac{x^n}{n!} e^{-\xi} \right| \leq \frac{1}{n!} e$ ($\because |x| \leq 1$ and $e^{-\xi} \leq e$)

Given $|\text{error}| \leq 5 \times 10^{-4}$

$\therefore \frac{1}{n!} e \leq 5 \times 10^{-4} \Rightarrow n! \geq e \times \frac{10^4}{5}$ i.e, $n! \geq 2000e$

The inequality is satisfied for $n = 8$ A

(iii) Given $|E_4| \leq 5 \times 10^{-4}$ ($\because |E_4| \leq \frac{|x^4|}{24} e$)

$\therefore$ We choose x_1 such that $\frac{|x^4|}{24} e \leq 5 \times 10^{-4}$

or $|x|^4 \leq \frac{120}{e} \times 10^{-4} = 0.00441$

Hence $|x| \leq 0.2577$

$\Rightarrow x \in [-0.2577, 0.2577]$ A

Normalized floating-point representation or normalized scientific notation

In the decimal system, any real no. (other than 0) can be represented in normalized floating point form as

$$x = \pm 0.d_1 d_2 d_3 \cdots \times 10^n$$

where $d_1 \neq 0$ and n is an integer (positive, negative or zero). The numbers $d_1, d_2, \cdots$ are the decimal digits $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$.

Alternatively, any real no. (other than 0) can be represented in normalized floating point decimal form as

$$x = \pm r \times 10^n \quad \left(\frac{1}{10} \leq r < 1\right)$$

Here the number r is called the normalized mantissa and integer n the exponent.

In the binary system, if $x \neq 0$, it can be written as

$$x = \pm q \times 2^m \quad \left(\frac{1}{2} \leq q < 1\right), \; (-126 \leq m \leq 127)$$

where m is any integer

The mantissa q would be expressed as a sequence of zeros or ones in the form $(0.b_1 b_2 b_3 \cdots)_2$ where $b_1 \neq 0$. Hence $b_1 = 1$ and then necessarily $q \geq \frac{1}{2}$.

Note that every computer has only a finite word length and a finite total capacity, so only numbers with a finite number of digits can be represented.

- A number is alloted only one word of storage in the single precision mode (two or more words in double or extended precision).

In either case, the degree of precision is strictly limited.

- The most real numbers cannot be represented exactly in a computer.
- The real numbers that are representable in a computer are called its machine numbers.
- A number that has a terminating expansion in one base may have a nonterminating expansion in another.

For ex: $\frac{1}{10} = (0.1)_{10} = (0.063146314631463146314\ldots)_8$

$= (0.0001100110011001100110011\ldots)_2$

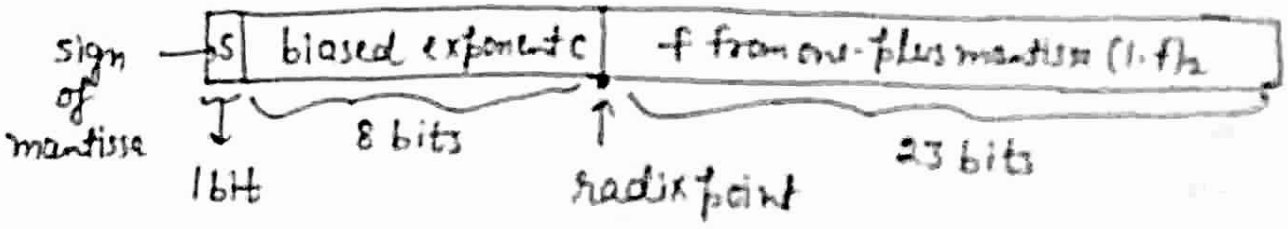
0.1 × 8 = 0.8
0.8 × 8 = 6.4
0.4 × 8 = 3.2
0.2 × 8 = 1.6
0.6 × 8 = 4.8
0.8 × 8 = 6.4

Single-Precision Floating-Point Form:

A machine number in standard single-precision floating-point form corresponds to

$$(-1)^s \times 2^{c-127} \times (1.f)_2 \quad , -126 \leq c-127 \leq 127$$

The leftmost bit is used for the sign of the mantissa, where s = 0 corresponds to + and s = 1 corresponds to −. The next eight bits are used to represent the number c in the exponent of 2^{c-127}, which is interpreted as an excess-127 code and the last 23 bits represent f from the fractional part of the mantissa in the 1-plus form: $(1.f)_2$. Each floating point single precision word is partitioned as



- **Note:** In the normalized representation of a nonzero floating-point number, the first bit in the mantissa is always 1 so that this bit does not have to be stored. This can be accomplished by shifting the binary point to a "1-plus" form $(1.f)_2$. The mantissa is the rightmost 23 bits and contains f with an understood binary point (radix point). So the mantissa (significand) actually corresponds to 24 binary digits since there is a hidden bit. (An important exception is the number ± 0)

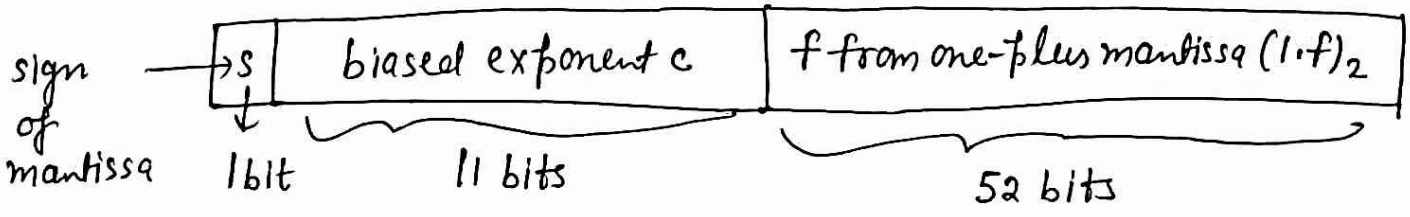
Double-Precision Floating-Point Form:

When more precision is needed, double precision can be used, in which case each double precision floating-point number is stored in two computer words in memory.

A machine number in standard double-precision floating-point form corresponds to

$$(-1)^s \times 2^{c-1023} \times (1.f)_2 \quad , \quad -1022 \leq c-1023 \leq 1023$$

which can be partitioned as



Que Determine the ~~single-precision~~ machine representation of the decimal number -52.234375 in both single precision and double precision.

Sol First we convert the integral part to binary, we have

$$(52.)_{10} = (64.)_8 = (110\ 100.)_2$$

8	52	
8	6	4 ↑
	0	6

Now, we convert the fractional part to binary

$(0.234375)_{10}$

$= (0.17)_8$

$= (0.001111)_2$

	0.234375
×	8
	1.875000
×	8
	7.000000

Note:

Binary:	000	001	010	011
Octal:	0	1	2	3
Binary:	**100**	**101**	**110**	**111**
Octal:	**4**	**5**	**6**	**7**

Now, $(52.234375)_{10}$

$= (110\ 100.001111)_2$

$= (1.101000\ 011\ 110)_2 \times 2^5$

is the corresponding one-plus form in base 2, and $(.101\ 000\ 011\ 110)_2$ is the stored mantissa.

Next the exponent is $(5)_{10}$ and since $c - 127 = 5$, we immediately see that $c = 132$ and $(132)_{10} = (204)_8$

8	132	
8	16	4 ↑
8	2	0
	0	2

$\therefore (10\ 000\ 100)_2$

is the stored exponent.

Thus, the single-precision machine representation of -52.234375 is

$$[1\ 10\ 000\ 100\ 101\ 000\ 011\ 110\ 000\ 000\ 000\ 00]_2$$

$= [1100\ 0010\ 0101\ 0000\ 1111\ 0000\ 0000\ 0000]_2$

$= [C250F000]_{16}$

Note

Binary :	0000	0001	0010	0011	0100	0101
Hexadecimal:	0	1	2	3	4	5
Binary :	6	7	8	9	A	B
Hexadecimal:	0110	0111	1000	1001	1010	1011
Binary :	1100	1101	1110	1111		
Hexadecimal:	C	D	E	F		

In double precision, for the exponent $(5)_{10}$, we let

$c - 1023 = 5 \Rightarrow c = 1028$

and we have $(1028)_{10} = (2004)_8$

$= (10\ 000\ 000\ 100)_2$

which is the store exponent.

8 | 1028		
8 | 128	4	↑
8 | 16	0	
8 | 2	0	
0	2	

Thus, the double-precision machine representation of -52.234375 is

$[1\ 10\ 000\ 000\ 100\ 101\ 000\ 011\ 110\ \underbrace{000\cdots00}_{40\ 0's}]_2$

$= [1100\ 0000\ 0100\ 1010\ 0001\ 1110\ \underbrace{0000\cdots0000}_{40\ 0's}]_2$

$= [C04A1E\underbrace{0000000000}_{10\ 0's}]_{16}$ A

Que Determine the decimal numbers that correspond to these machine words:

$[45DE4000]_{16}$ $[BA390000]_{16}$

Sol (i) $[45DE4000]_{16} = [0100\ 0101\ 1101\ 1110\ 0100\ 0000\ 0000\ 0000]_2$

0 → It shows sign is +ive; 10001011 → biased exponent; 11011110010000000000000 → f from one-plus mantissa $(1.f)_2$

Stored exponent $= (10001011)_2 = (1\times2^0+1\times2^1+1\times2^3+1\times2^7)_{10}$
$= (1+2+8+128)_{10} = (139)_{10}$

$\therefore c = 139 \Rightarrow 2^{c-127} = 2^{139-127} = 2^{12}$

Now, the mantissa is +ive and represents the number

$(1.101\ 111\ 001)_2 \times 2^{12} = (1101\ 1110\ 0100\ 00)_2$

$= 2^3+2^6+2^7+2^8+2^9+2^{11}+2^{12} = 2^3(1+2^3+2^4)+2^8(1+2+2^3+2^4)$

$= 8\times(1+8+16)+256(1+2+8+16)$

$= 8\times25+256\times27 = 200+6912 = (7112)_{10}$ ∆

or first we can write in octal form as $(213)_8 = 2\times8^2+8+3 = 128+8+3 = (139)_{10}$

(ii) $[BA390000]_{16}$

$= [1011\ 1010\ 0011\ 1001\ 0000\ 0000\ 0000\ 0000]_2$

1 → sign is -ve; 01110100 → biased exponent; 01110010000000000000000 → f from one-plus mantissa $(1.f)_2$

Stored exponent $= (01110100)_2 = (164)_8 = (1\times8^2+6\times8+4)_{10}$
$= (64+48+4)_{10} = (116)_{10}$

$\therefore c = 116 \Rightarrow 2^{c-127} = 2^{116-127} = 2^{-11}$

Now, the mantissa is negative and represents the number

$-(1.011\ 1001)_2 \times 2^{-11} = -(0.000\ 000\ 000\ 010\ 111\ 001)_2$

$= -(0.000271)_8 = -(2\times8^{-4}+7\times8^{-5}+8^{-6})_{10}$

$= -8^{-6}(1+56+128)$

$= -\frac{185}{8^6} = \frac{-185}{262144}$

$\approx -7.0571899\times10^{-4}$ ∆

Note:

(1) Machine epsilon: When we are using single precision, the binary machine floating-point number $\varepsilon = 2^{-23}$ is called the machine epsilon. It is the smallest positive machine number ε such that $1+\varepsilon \neq 1$. When we are using double precision, the machine epsilon is 2^{-52}.

Since $2^{-23} \approx 1.2 \times 10^{-7}$ and $2^{-52} \approx 2.2 \times 10^{-16}$

∴ Approximately 6 significant decimal digits of accuracy may be obtained in single precision while approximately 15 significant decimal digits of accuracy may be obtained in double precision.

(2) Single precision on a 64-bit computer is comparable to double precision on a 32-bit computer, whereas double precision on a 64-bit computer gives four times the precision available on a 32-bit computer.

Floating-point machine number:

Suppose that we are working with a five-place decimal machine and wish to add numbers.

For ex: $x = 0.37218 \times 10^4$ and $y = 0.71422 \times 10^{-1}$ are two normalized floating point machine numbers. Then we can find $x+y$ as

$$x = 0.3721800000 \times 10^4$$
$$y = 0.0000071422 \times 10^4$$
$$x+y = 0.3721871422 \times 10^4$$

(adjust the exponent of smaller number so that both exponents are same to add them)

The nearest machine number is $z = 0.37219 \times 10^4$

$\therefore$ Relative error (involved in this addition)

$$= \frac{|(x+y)-z|}{|x+y|} = \frac{0.0000028578 \times 10^4}{0.3721871422 \times 10^4} \approx 0.77 \times 10^{-5}$$

To facilitate the analysis of such errors, we introduce the notation $fl(x)$.

Notation $fl(x)$: The notation $fl(x)$ is used to denote the floating point machine number that corresponds to the real number x.

Note: $fl(x) = x(1+\delta)$, $|\delta| \leq 2^{-24}$

where δ is the relative error.

Computer Arithmetic: Let the symbol $\odot$ denote any one of the arithmetic operations $+, -, \times$ or $\div$. Suppose a 32-bit word length computer has been designed so that whenever two machine numbers x and y are to be combined automatically, the computer produces $fl(x \odot y)$ instead of $x \odot y$. We can imagine that $x \odot y$ is first correctly formed, then normalized, any finally rounded to become a machine number.

$\therefore$ $fl(x \odot y) = (x \odot y)(1+\delta)$; $|\delta| \leq 2^{-24}$

Hence $fl(x \pm y) = (x \pm y)(1+\delta)$ where δ is the relative error

$fl(xy) = xy(1+\delta)$

$fl\left(\frac{x}{y}\right) = \left(\frac{x}{y}\right)(1+\delta)$

where $-2^{-24} \leq \delta \leq 2^{24}$

Que If x, y and z are machine numbers in a 32-bit word-length computer, what upper bound can be given for the relative roundoff error in computing $z(x+y)$?

Sol In the computer, the calculation of $(x+y)$ would be done first and produces the machine number $fl(x+y)$, which differ from $x+y$ because of roundoff.

$$\therefore fl(x+y) = (x+y)(1+\delta_1) \quad (\text{for some } \delta_1 \text{ such that } |\delta_1| \leq 2^{-24})$$

When z multiplies the machine number $fl(x+y)$, the result is the machine number $fl[z\,fl(x+y)]$ because z is also a machine number.

$$\therefore fl[z\,fl(x+y)] = z\,fl(x+y)(1+\delta_2) \quad (\text{for some } \delta_2 \text{ such that } |\delta_2| \leq 2^{-24})$$

Hence
$$\begin{aligned} fl[z\,fl(x+y)] &= z(x+y)(1+\delta_1)(1+\delta_2) \\ &= z(x+y)(1+\delta_1+\delta_2+\delta_1\delta_2) \\ &\approx z(x+y)(1+\delta_1+\delta_2) \\ &\qquad (\because |\delta_1\delta_2| \leq 2^{-48} \text{ and so we ignore it}) \\ &= z(x+y)(1+\delta) \quad (\text{where } \delta = \delta_1+\delta_2) \end{aligned}$$

Now, relative roundoff error in computing $z(x+y)$

$$= |\delta|$$
$$= |\delta_1+\delta_2| \leq |\delta_1|+|\delta_2| \leq 2^{-24}+2^{-24} = 2^{-23}$$

Hence $|\delta| \leq 2^{-23}$ A

Que Show by an example that in computer arithmetic $a+(b+c)$ may differ from $(a+b)+c$.

Sol Let $a = 0.345$, $b = 0.245 \times 10^{-3}$ and $c = 0.432 \times 10^{-3}$ and we are using 3-digit rounding.

Here $a = 0.345 \times 10^{0}$

$b = 0.000245 \times 10^{0}$

$c = 0.000432 \times 10^{0}$

Now, $b+c = 0.000677 \times 10^{0} = 0.677 \times 10^{-3}$ ($\because$ write numbers in normalized floating form)

$\therefore a+(b+c) = (0.345+0.000677) \times 10^{0}$

$= 0.345677 \times 10^{0} \approx \boxed{0.346 \times 10^{0}}$

Also, $a+b = 0.345245 \times 10^{0} = 0.345 \times 10^{0}$

$\therefore (a+b)+c = (0.345+0.000432) \times 10^{0}$

$= 0.345432 \times 10^{0} \approx \boxed{0.345 \times 10^{0}}$

Hence, $a+(b+c) \neq (a+b)+c$

Que If x and y are real numbers within the range of a 32-bit word-length computer and if xy is also within the range, what relative error can there be in the machine computation of xy?

Sol Machine produces $fl[fl(x)fl(y)]$.

Now, $fl(x) = x(1+\delta_1)$ for some δ_1 such that $|\delta_1| \leq 2^{-24}$

$fl(y) = y(1+\delta_2)$ " " δ_2 " " $|\delta_2| \leq 2^{-24}$

$\therefore fl[fl(x)fl(y)] = fl(x)fl(y).(1+\delta_3)$ for some δ_3 such that $|\delta_3| \leq 2^{-24}$

$= xy(1+\delta_1)(1+\delta_2)(1+\delta_3)$

$\approx xy(1+\delta_1+\delta_2+\delta_3)$ (neglecting other terms as they are very small and will not affect the calculation)

$= xy(1+\delta)$; $\delta = \delta_1+\delta_2+\delta_3$

$\therefore$ relative error $= |\delta| \leq |\delta_1|+|\delta_2|+|\delta_3| = 3.2^{-24}$ A

Loss of Significance: Loss of significance occurs in numerical calculations when too many significant digits cancel.

Significant digits Suppose that x is a real number expressed in normalized scientific notation in the decimal system

$$x = \pm r \times 10^n \quad \left(\frac{1}{10} \leq r < 1\right)$$

For Ex: $x = 0.3721498 \times 10^5$

The digits 3, 7, 2, 1, 4, 9, 8 used to express r do not have the same significance because they represent different powers of 10.

Here 3 is the most significant digit
and 8 is the least significant digit

Note that the significance of the digits diminishes from left to right.

Que If $x = 0.3721448693$
$y = 0.3720214371$, then what is the relative error in the computation of $x-y$ in a computer that has five decimal digits of accuracy?

Sol Exact value of $x-y = 0.3721448693 - 0.3720214371$
$= 0.0001234322$

Approximate value of $x-y$ (with five decimal digits of accuracy)
$= 0.37214 - 0.37202 = 0.00012$

$$\therefore \text{Relative error involved} = \left|\frac{\text{Exact value} - \text{Approximate value}}{\text{Exact value}}\right|$$

$$= \left|\frac{0.0001234322 - 0.00012}{0.0001234322}\right| = \frac{0.0000034322}{0.0001234322}$$

$\approx 3 \times 10^{-2}$ (which is quite large as by the coarsest estimates it cannot exceed $\frac{1}{2} \times 10^{1-5} = \frac{1}{2} \times 10^{-4}$)

Loss of precision theorem: Let x and y be normalized floating-point machine numbers, when $x > y > 0$. If

$2^{-p} \leq 1 - \frac{y}{x} \leq 2^{-q}$ for some positive integers p and q, then at most p and at least q significant binary bits are lost in the subtraction $x - y$.

Que In the subtraction $37.593621 - 37.584216$, how many bits of significance will be lost?

Sol Let $x = 37.593621$

and $y = 37.584216$

Then $1 - \frac{y}{x} = 0.0002501754$

This lies between $2^{-12} = 0.000244$ and $2^{-11} = 0.000488$.

Hence, at least 11 but not more than 12 bits are lost.

Remark: To avoid loss of significance in subtraction, one may be able to reformulate the expression using rationalizing, series expansions, or mathematical identities.

Ex: Consider the function

$$f(x) = \sqrt{x^2+1} - 1$$

whose value may be required for x near zero.

Since $\sqrt{x^2+1} \approx 1$ when $x \approx 0$, we see that there is a potential loss of significance in the subtraction.

$\therefore$ We can rationalized the numerator to avoid loss of significance as

$$f(x) = (\sqrt{x^2+1} - 1) \times \frac{(\sqrt{x^2+1}+1)}{(\sqrt{x^2+1}+1)}$$

$$= \frac{(x^2+1)-1}{\sqrt{x^2+1}+1} = \frac{x^2}{\sqrt{x^2+1}+1} \quad (\because \text{It removes subtraction})$$

Que How can accurate values of the function

$$f(x) = e^x - e^{-2x} \qquad \text{——— (1)}$$

be computed in the vicinity of $x = 0$?

Sol Since e^x and e^{-2x} are both equal to 1 when $x = 0$, therefore there is a loss of significance in the subtraction when x is closed to zero.

One cure of this problem is to use the Taylor series as

$$f(x) = \left(1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots\right) - \left(1 - 2x + \frac{4x^2}{2!} - \frac{8x^3}{3!} + \cdots\right)$$

$$= 3x - \frac{3}{2}x^2 + \frac{3}{2}x^3 - \cdots \qquad \text{——— (2)}$$

Extra: Find the range in which series (2) should be used and the range in which formula (1) can be used.

Sol Using the Theorem on Loss of Precision, we see that the loss of bits in the subtraction of formula (1) can be limited to at most 1 bit by restricting x so that

$$2^{-1} \le 1 - \frac{e^{-2x}}{e^x} \Rightarrow \frac{1}{2} \le 1 - \frac{1}{e^{3x}} \Rightarrow \frac{1}{e^{3x}} \le \frac{1}{2} \quad (\text{when } x > 0)$$

$$\Rightarrow e^{3x} \ge 2$$

$$\Rightarrow 3x \ge \ln 2$$

$$\Rightarrow x \ge \frac{1}{3}\ln 2 = 0.23105$$

Similarly when $x < 0$, then for $x \le -0.23105$ at most 1 bit is lost.

Hence the series (2) should be used for $|x| \leq 0.23105$ and for $|x| > 0.23105$, formula (1) can be used.

Range Reduction:

Another cause of loss of significant figures is the evaluation of various library functions with large arguments. For ex:-

A basic property of the function $\sin x$ is its periodicity;

$$\sin x = \sin(x + 2n\pi)$$ for all real values of x and for all integer values of n.

Because of this relationship, we need to know only the values of $\sin x$ in some fixed interval of length 2π to compute $\sin x$ for arbitrary x. This property can be used in the computer evaluation of $\sin x$ and is called range reduction.

Que For $\sin x$, how many binary bits of significance are lost in range reduction to the interval $[0, 2\pi)$?

Sol Given an argument $x > 2\pi$, we find an integer n that satisfies $0 \leq x - 2n\pi < 2\pi$

Then in evaluating elementary trigonometric functions, we use $f(x) = f(x - 2n\pi)$

In the subtraction $x - 2n\pi$, there is a loss of significance.

By the Loss of Precision theorem, at least q bits are lost if

$$1 - \frac{2n\pi}{x} \leq 2^{-q}$$

Since $$1 - \frac{2n\pi}{x} = \frac{x - 2n\pi}{x} < \frac{2\pi}{x}$$

we conclude that at least q bits are lost if $\frac{2\pi}{x} \leq 2^{-q}$

or $2^q \leq \frac{x}{2\pi}$ A