Floating Point Representation And Computer Arithmetic.

Since the base for a computer is (almost always) 2 rather than 10, a number N is represented as

$$N = 0.d_1 d_2 d_3 \ldots d_p \times 2^E.$$

$$= \left[(d_1)\frac{1}{2} + (d_2)\frac{1}{2^2} + (d_3)\frac{1}{2^3} + \cdots + (d_p)\frac{1}{2^p}\right] \times 2^E.$$

where each digit $d_1, d_2, d_3, \ldots, d_p$ is either 0 or 1.

The decimal part $0.d_1 d_2 d_3 \ldots d_p$ is called the mantissa.

The precision of the number depends on the number of digits in the mantissa, i.e., the value of p. The range of numbers that can be represented depends on the range of values for the exponent E; we indicate that range as $L \leq E \leq U$.

If we assume that binary floating-point number system is normalised, so that the leading digit $d_1 = 0$ (unless, $N = 0$), then the representation of each number is unique, no digits are wasted on leading zeroes, and there is no need to store the value of d_1 (since it is known to be one).

Binary Value	Normalised As	Exponent
1101.101	0.1101101	4
.00101	0.101	−2
1.0001	0.10001	1
10000011.0	0.10000011	8

The floating-point binary value 1101.101 is normalised as 0.1101101 by moving the decimal point 4 positions to the left, and multiplying by 2^4.

The number of positive values that can be represented exactly in a normalised binary floating-point system is $V = 2^{P-1}(U-L+1)$; allowing for representation of positive and negative values, along with zero, gives the total numbers of "machine numbers" (that is, numbers that can be represented exactly) as $2V+1$.

In adding or subtracting floating-point numbers, the exponents must agree. If they do not, the mantissa of the smaller (magnitude) must be shifted, and some digits must be lost. In floating-point multiplication, the mantissas are multiplied and the exponents added. The additional digits in the mantissa are rounded.

Example. Floating Point Addition and multiplication

For example, consider a small base-10 system, with four digits of precision. Let $a = 0.4321 \times 10^2 = 43.21$ and $b = 0.1234 \times 10^{-1} = 0.01234$.

Since exponent of a is 2 and that of b is -1.

Therefore, to add, b must be shifted to the form

$$b = 0.0001234 \times 10^2 \quad [\text{Exponent} = 2]$$

since b is smaller in ~~magnt~~ magnitude.

Adding gives $a+b = 0.432234 \times 10^2$, which rounds to $a+b = 0.4322 \times 10^2$. Thus only the first digit of b has any effect on the result.

To multiply, we have $a \times b = 0.05332114 \times 10^{2+(-1)}$ which rounds to $a \times b = 0.0533 \times 10^1$

Example. Conversion from Decimal to Binary.

The decimal fractions that can be represented exactly with $p = 3$ digits and $E = -1$ or $E = 0$ are total 8 in number as $U = 0$, $L = -1$ and $p = 3$.

Thus $V = 2^{p-1}(U - L + 1) = 2^{3-1}(0-(-1)+1)$

$= 2^2(2) = 8$

The leading decimal is 1 for each number, since we are assuming a normalised floating-point system.

Exponent	Binary	Decimal Expansion	Decimal
-1	0.0100_2	$(1)(\frac{1}{4}) + (0)(1/8) + (0)(1/16)$	0.25
-1	0.0101_2	$(1)(1/4) + (0)(1/8) + (1)(1/16)$	0.3125
-1	0.0110	$(1)(1/4) + (1)(1/8) + (0)(1/16)$	0.375
-1	0.0111	$(1)(1/4) + (1)(1/8) + (1)(1/16)$	0.4375
0	0.1000	$(1)(1/2) + (0)(1/4) + (0)(1/8)$	0.5
0	0.1010_2	$(1)(1/2) + (0)(1/4) + (1)(1/8)$	0.625

Exponent	Binary Decimal	Expansion	Decimal
0	0.1100_2	$(1)(1/2)+(1)(1/4)+(0)(1/8)$	0.75
0	0.1110_2	$(1)(1/2)+(1)(1/4)+(1)(1/8)$	0.875

⇒ It is useful to note that the numbers that can be represented exactly are not evenly distributed between the largest and smallest such numbers, but rather are evenly distributed between successive powers of the base.

Error :→

Error arises because of :→

1. Simplifying assumptions made in modelling the original problem.
2. Errors arising from data collection
3. Errors that arise because computers do not represent most numbers in an exact form.

Measuring Error :→

If x is our approximate result, and the exact (but usually unknown) result is denoted by x^*, then the error in using the approximate result is

$$\text{Error}(x) = x^* - x = \text{Actual value} - \text{approximate value}$$

$$\text{Absolute Error} = |x^* - x| = |x - x^*|$$

However, especially for problems in which # the magnitude of the true value may be very large, or very small, the relative error may be more important than the actual error.

$$\text{Rel. Error} = \frac{x^* - x}{x^*}$$

In computing the relative error, the approximate value is often used in the denominator in place of unknown true value x^*, then the relative error will be.

$$\text{Rel. Error} = \frac{x^* - x}{x}$$

Significant Digits :→

The number x is said to approximate x^* to 't' significant digits if t is the largest non-negative integer for which

$$\left|\frac{x - x^*}{x}\right| < 5 \times 10^{-t}$$

Errors From Inexact Representation.

The error introduced by shortening a number that has more digits that can be represented by the available floating-point system is known as round-off error. There are two basic approaches. The simplest is to chop the number, discarding any digits beyond what the system can accomodate. The other is to round the number; the result depends on the value of the first digit to be discarded. If the system allows for n digits, rounding produces the same result as chopping if the $(n+1)^{st}$ digit is 0, 1, 2, 3, or 4. If the $(n+1)^{st}$ digit is 6, 7, 8, or 9, the n^{th} digit is increased by 1. If the $(n+1)^{st}$ digit is 5, it is common to round so that the n^{th} digit is even rounding up about half the time. ⇒ The inaccuracies that occur from either rounding or chopping are known as round-off errors.

Example. Effect of Order of Operations.

As an example of round-off, consider the following addition problem: $0.99 + 0.0044 + 0.0042$

With exact arithmetic, the result is 0.9986, regardless of the order in which the addition is performed. However, if we have three digit arithmetic, and the operations are nested from left to right, we find that

$$(0.99 + 0.0044) + 0.0042 \approx 0.944 + 0.0042 \approx 0.998$$

On the other hand, if we change the nesting so that the small numbers are added together first, we get

$$0.99 + (0.0042 + 0.0044) = 0.99 + 0.0086 \approx 0.999$$

0.998 approximate the true solution $x^* = 0.9986$, to three significant digits, since

$$\left|\frac{x - x^*}{x}\right| = \left|\frac{0.998 - 0.9986}{0.998}\right| = 6.012 \times 10^{-4} < 5 \times 10^{-3}$$

On the other hand, 0.999 approximates the true solution, $x^* = 0.9986$, to four significant digits, since

$$\left|\frac{0.999 - 0.9986}{0.999}\right| = 4.004 \times 10^{-4} < 5 \times 10^{-4}$$