Error analysis - 1

20 January 2026 12:09

Error Analysis

Exact Solution is approximated using numerical schemes, which incur an error called mathematical error (ME). Therefore,

Exact solution = Approximate Solution + Mathematical Error.

A finite digit arithmetic machine machine introduces additional error when a numerical scheme is implemented due to approximation of numbers using finite digits. This error is called arithmetic error and the output is called numerical solution.

AS = Numerical Solution + Arithmetic Error

$$ES = NS + \boxed{AE + ME} \rightarrow TE$$

Floating point representation

Let $\beta \in \mathbb{N}$ and $\beta \geq 2$. A real number can be represented exactly in base β as

$$(-1)^s \times (0.d_1 d_2 \cdots d_n d_{n+1} \cdots)_\beta \times \beta^e$$

where, $d_i \in \{0, 1, \cdots, \beta-1\}$ with $d_1 \neq 0$ OR $d_1 = \cdots = d_n = \cdots = 0$,

$s = 0$ or 1 and the appropriate integer e is called the exponent.

Here,

$$0.d_1 d_2 \cdots d_n d_{n+1} \cdots = \frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \cdots + \frac{d_n}{\beta^n} + \frac{d_{n+1}}{\beta^{n+1}} + \cdots$$

is called the mantissa (β-fraction), s - sign, β - radix. This representation is called floating point representation.

Floating point approximation

An n-digit floating-point number in base β is of the form

$$(-1)^s \times (0.d_1 d_2 \cdots d_n)_\beta \times \beta^e$$

$d_1 \neq 0$ or $d_1 = d_2 = \cdots = d_n = 0$, $s = 0$ or $s = 1$ and an appropriate exponent e.

Underflow and overflow of Memory

In every computing device, the exponent e has an upper bound and a lower bound.

That is $m < e < M$.

If in a calculation, $e > M$, then memory overflow occurs and if $e < m$, the memory underflow occurs.