

2 Truncation errors in floating point representation

When a real value x is stored using its floating point representation $fl(x)$, truncation error occurs. This is because of the need to represent an infinite number of real values using a finite number of bits.

Example: the floating point representation of $(0.1)_{10}$

$$(0.1)_{10} = (0.0001\ 1001\ 1001\ 1001\ 1001\ 1001\ 1001\ 1001\ \dots)_2$$

2.1 Truncation errors

Assume: we have t number of bits to represent m (or equivalently $t - 1$ bits for f). In IEEE 32-bit format, $t - 1 = 23$, or $t = 24$.



Figure 14: Truncation error

With a finite number of bits, there are two ways to approximate a number that cannot be represented exactly in floating point format.

Consider real value x which cannot be represented exactly and falls between two floating point representations $fl(x_1)$ and $fl(x_2)$.

Chopping:

$$fl(x) = fl(x_1)$$

— Ignore all bits beyond the $(t - 1)$ th one in f (or t th one in m).

Rounding:

$$fl(x) = \begin{cases} fl(x_1), & \text{if } x - fl(x_1) \leq fl(x_2) - x \\ fl(x_2), & \text{otherwise.} \end{cases}$$

— Select the closest representable floating point number.

Example: $x > 0$,

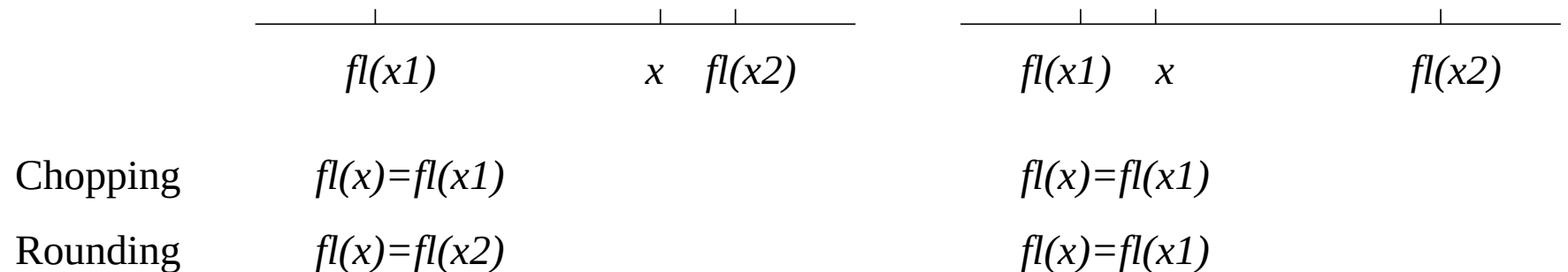


Figure 15: Example of using chopping and rounding

As x increases, $fl(x_2) - fl(x_1)$ increases, then the truncation error increases.

Absolute error:

$$\text{Absolute error} \triangleq |x - fl(x)|$$

is the difference between the actual value and its floating point representation.

Relative error:

$$\text{Relative error} \triangleq \frac{|x - fl(x)|}{|x|}$$

is the error with respect to the value of the number to be represented.

2.2 Bound on the errors ($b = 2$)

Consider chopping, $x > 0$.

The real value x can be represented exactly with an infinite number of bits as

$$x = 1. \underbrace{XX \dots XX}_{t-1 \text{ bits}} \underbrace{XX \dots XX}_{\infty \text{ bits}} \times b^e$$

The floating point representation $fl(x)$ is

$$fl(x) = 1. \underbrace{XX \dots XX}_{t-1 \text{ bits}} \times b^e$$

The absolute error is maximum when

$$x = 1. \underbrace{XX \dots XX}_{t-1 \text{ bits}} \underbrace{11 \dots 11}_{\infty \text{ bits}} \times b^e$$

Then, the absolute error is bounded by

$$|x - fl(x)| < \sum_{i=t}^{\infty} b^{-i} \times b^e < b^{-t+1} \times b^e$$

and the relative error is bounded by

$$\frac{|x - fl(x)|}{|x|} < \frac{b^{1-t} \times b^e}{|x|}$$

Because $|x| > 1.00 \dots 00 \times b^e = b^e$,

$$\frac{|x - fl(x)|}{|x|} < \frac{b^{1-t} \times b^e}{b^e} = b^{1-t}$$

For binary representation, $b = 2$,

$$\frac{|x - fl(x)|}{|x|} < 2^{1-t}$$

which is the bound on relative errors for chopping.

For chopping,

$$|x - fl(x)| \leq [fl(x_2) - fl(x_1)]$$

For rounding: the maximum truncation error occurs when x is in the middle of the interval between $fl(x_1)$ and $fl(x_2)$. Then

$$|x - fl(x)| \leq \frac{1}{2}[fl(x_2) - fl(x_1)]$$

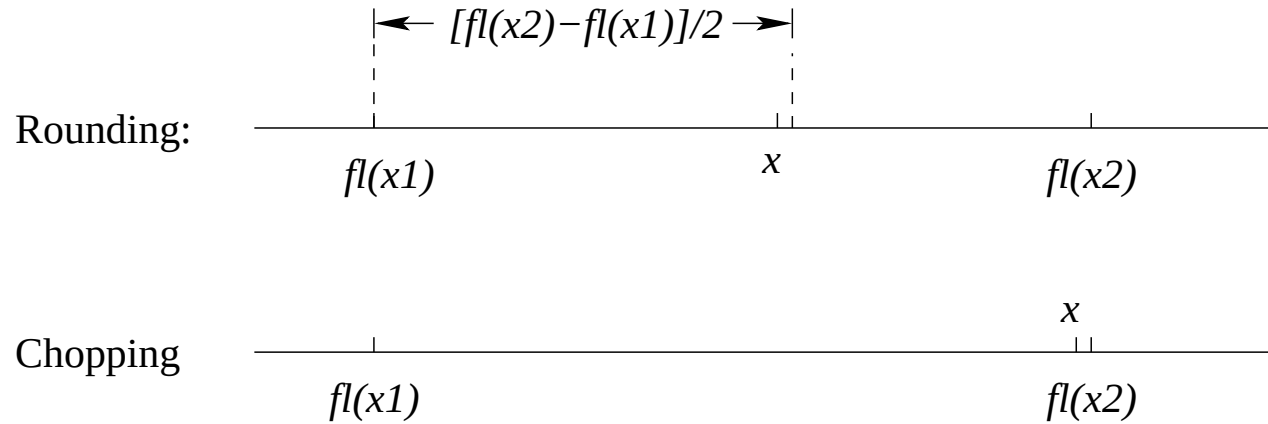


Figure 16: Illustration of error bounds

The bound on absolute errors for rounding is

$$|x - fl(x)| \leq \frac{2^{-t+1} \cdot 2^e}{2} = 2^{-t} \cdot 2^e$$

The bound on relative errors for rounding is

$$\frac{|x - fl(x)|}{|x|} \leq \frac{2^{-t+1}}{2} = 2^{-t}$$

Machine precision: Define machine precision, ϵ_{mach} , as the maximum relative error. Then

$$\epsilon_{mach} = \begin{cases} 2^{1-t}, & \text{for chopping} \\ \frac{2^{1-t}}{2} = 2^{-t}, & \text{for rounding} \end{cases}$$

For IEEE standard, $t - 1 = 23$ or $t = 24$, $\epsilon_{mach} = 2^{-24} \approx 10^{-7}$ for rounding.

The machine precision ϵ_{mach} is also defined as the smallest number ϵ such that $fl(1 + \epsilon) > 1$, i.e., if $\epsilon < \epsilon_{mach}$, then $fl(1 + \epsilon) = fl(1)$.

(Prove that the two definitions are equivalent.)

Note: Difference between the machine precision, ϵ_{mach} , and the lower bound on the representable floating point numbers, L :

- ϵ_{mach} is determined by the number of bits in m
- L is determined by the number of bits in the exponent e .

2.3 Effects of truncation and machine precision

Example 1: Evaluate $y = (1 + \delta) - (1 - \delta)$, where $L < \delta < \epsilon_{mach}$. With rounding,

$$\begin{aligned} y &= fl(1 + \delta) - fl(1 - \delta) \\ &= 1 - 1 = 0. \end{aligned}$$

The correct answer should be 2δ .

When $\delta = 1.0 \times 2^{-25}$,

$$\begin{aligned} 1 + \delta &= 1.0 \times 2^0 + 1.0 \times 2^{-25} \\ &= (1.\underbrace{00 \cdots 0}_{24 \text{ 0's}}1)_2 \times 2^0 \\ fl(1 + \delta) &= 1 \end{aligned}$$

Example 2: Consider the infinite summation $y = \sum_{n=1}^{\infty} \frac{1}{n}$. In theory, the sum diverges as $n \rightarrow \infty$. But, using floating point operations (with finite number of bits), we get a finite output.

As n increases, the addition due to another $\frac{1}{n}$ does not change the output! That is,

$$\begin{aligned} y &= \sum_{n=1}^{\infty} \frac{1}{n} = \sum_{n=1}^{k-1} \frac{1}{n} + \sum_{n=k}^{\infty} \frac{1}{n} = \sum_{n=1}^{k-1} \frac{1}{n} \\ \frac{1}{k-1} &\geq \epsilon_{mach} \quad \text{and} \quad \frac{1}{k} < \epsilon_{mach} \end{aligned}$$

Example 3: Consider the evaluation of e^{-x} , $x > 0$, using

$$e^{-x} = \sum_{n=0}^{\infty} \frac{(-x)^n}{n!} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots$$

This may result in erroneous output due to cancellation. Alternative expressions are needed for evaluation with negative exponents.

2.4 Significant figures (digits)



Figure 17: Significant figures

The significant digits of a number are those that can be used with confidence. For example:

$$0.00453 = 4.53 \times 10^{-3} \text{ 3 significant digits}$$

$$0.004530 = 4.530 \times 10^{-3} \text{ 4 significant digits}$$

Example: $\pi = 3.1415926 \dots$

With 2 significant digits, $\pi \approx 3.1$

With 3 significant digits, $\pi \approx 3.14$

With 4 significant digits, $\pi \approx 3.142$

The number of significant digits is the number of certain digits plus one estimated digit.