MA633L-Numerical Analysis

Lecture 7 : Chopping, Rounding, Truncation Errors

Panchatcharam Mariappan[1]

[1]Associate Professor
Department of Mathematics and Statistics
IIT Tirupati, Tirupati

January 17, 2025

Machine Epsilon

Let p denote the number of bits in the significand or fraction bits. Then, the machine epsilon or the interval machine epsilon is

$$\varepsilon = 2^{-(p-1)}$$

whereas the rounding machine epsilon is given by

$$\epsilon_R = 2^{-p}$$

Data Type	p	ε	ε_R
Single (float)	24	1.19×10^{-7}	5.96×10^{-8}
Double (double)	53	2.22×10^{-16}	1.11×10^{-16}
Long Double (80-bit)	64	1.08×10^{-19}	5.4×10^{-20}
Long Double (128-bit)	113	1.93×10^{-34}	9.63×10^{-35}

Table 1: Machine epsilon values for common floating-point types.

Overflow

- Without considering the sign of the digit, single, double and quadruple precision floating points respectively can represent a finitely many values in the interval $[SP_{min}, SP_{max}]$, $[DP_{min}, DP_{max}]$ and $[QP_{min}, QP_{max}]$.
- In a computation, if a number x outside these interval occurs, then either **underflow** or **overflow** occurs.
- If x is a single precision result and $|x| \geq SP_{max}$, then it is called **overflow.**
- During the overflow, a few computers cease to function, whereas standard codes are written to avoid overflow.

The same argument applies when x is double or quadruple precision with their respective min and max values

Underflow

- If $|x| \leq SP_{min}$, then it is called **underflow**.
- For underflow, usually $x = 0$ is assigned and the computation continues.

The same argument applies when x is double or quadruple precision with their respective min and max values.

Machine Epsilon, Smallest/Largest Number

Aspect	Machine Epsilon	Smallest Number	Largest Number
Definition	Smallest difference between 1 and the next representable number	Smallest normalized positive number that can be represented	Largest normalized positive number that can be represented
Represents	Precision of the floating -point system	The lower bound of representable numbers	The upper bound of representable numbers
Significance	Determines how accurately numbers can be stored and computation can be performed	Determines the smallest non-zero value the system can handle without underflowing	Defines the largest value that can be represented before overflow
Range	Typically a very small number, much smaller than both the smallest and largest numbers	The lower limit of the system's number range	The upper limit of the system's number range
Example	$\approx 2.22E-16$	$\approx 2.22E-308$	$\approx 1.8E308$

Chopping

Chopping

Chopoff errors occurs as digital computers cannot represent some quantities exactly. They are important to engineering and scientific problems solving because they can lead to erroneous results. In some cases, it can lead to unstable results, said to be ill-conditioned. When a has a floating point base-10 system representation,

$$a = \pm m \times 10^n = \pm 0.d_1 d_2 \cdots d_k d_{k+1} d_{k+2} \cdots \times 10^n \tag{1}$$

and if we chop off the digits from d_{k+1}, it produces

$$a_C = \pm \overline{m} \times 10^n = \pm 0.d_1 d_2 \cdots d_k \times 10^n \tag{2}$$

Chopping

Chopoff Rule: For chopping, we simply chop off all but the first k digits, to obtain a_C. Since, we are discarding all decimals from some decimal on, it is also called as chopping error.

$$
\begin{aligned}
E_{tabs} &= |a - a_C| \approx |m - \overline{m}| \times 10^n \\
&= |0.d_1 d_2 \cdots d_k d_{k+1} d_{k+2} \cdots - 0.d_1 d_2 \cdots d_k| \times 10^n \\
&= |0.d_{k+1} d_{k+2} \cdots| \times 10^{n-k} \leq 10^{n-k} \\
\epsilon_t &= \left| \frac{a - a_C}{a} \right| \approx \left| \frac{m - \overline{m}}{m} \right| = \frac{|0.d_{k+1} d_{k+2} \cdots| \times 10^{n-k}}{|0.d_1 d_2 \cdots d_k d_{k+1} d_{k+2} \cdots| \times 10^n} \\
&\leq \frac{10^{n-k}}{|0.d_1 d_2 \cdots d_k d_{k+1} d_{k+2} \cdots| \times 10^n} \leq \frac{1}{0.1} 10^{-k} = 10^{1-k}
\end{aligned}
$$

Chopping

The last step is obtained as the numerator is bounded by 1, $d_1 \neq 0$ and the minimal value of the denominator is 0.1

Definition 1 (Chopoff Error)
Let a_C denote floating point approximation of a obtained by chopping the first k-digits, then the chopoff rule gives the relative error as

$$\epsilon_t = \left|\frac{a - a_C}{a}\right| \approx \left|\frac{m - \overline{m}}{m}\right| \leq 10^{1-k}.$$

The right side $u = 10^{1-k}$ is called the chopping unit.

Roundoff

Roundoff

Similar to chopping, we can also, obtain the rounding off as follows:

$$a_R = \pm\overline{m} \times 10^n = \pm 0.\delta_1\delta_2\cdots\delta_k \times 10^n \tag{3}$$

Here,

$$\delta_k = \begin{cases} d_k + 1 & \text{if } d_{k+1} \geq 5 \\ d_k & \text{if } d_{k+1} < 5 \end{cases}$$

$$a_R = \begin{cases} \pm(0.d_1d_2\cdots d_k \times 10^n + 10^{n-k}) & \text{if } d_{k+1} \geq 5 \\ \pm 0.d_1d_2\cdots d_k \times 10^n & \text{if } d_{k+1} < 5 \end{cases}$$

Roundoff

Roundoff Rule: For rounding, when $d_{k+1} \geq 5$, we add 1 to d_k and obtain δ_k and chop off the rest, to obtain a_R, we name it as round up. When $d_{k+1} < 5$, we simply chop off all but the first rest k digits, to obtain a_R, we name it as round down.

Roundoff

Example 2

In an Excel sheet, you can work with the following:

- ROUND(1.2535,1)=1.3
- ROUND(1.2535,2)=1.25
- ROUND(1.2535,3)=1.254
- ROUND(1.99999999,6)=2

Roundoff

Example 3

Find the five-digit (a) round and (b) chop off values of the irrational number π.

Solution: $\pi = 3.14159265...$

$$a = 0.314159265... \times 10^1$$

(a)

$$Roundup = 0.31416 \times 10^1 = 3.1416$$

(b)

$$Chop = 0.31415 \times 10^1 = 3.1415$$

Roundoff Error

If $d_{k+1} < 5$

$$
\begin{aligned}
E_{tabs} &= |a - a_R| \approx |m - \overline{m}| \times 10^n \\
&= |0.d_1 d_2 \cdots d_k d_{k+1} d_{k+2} \cdots - 0.d_1 d_2 \cdots d_k| \times 10^n \\
&= |0.d_{k+1} d_{k+2} \cdots| \times 10^{n-k} \leq 10^{n-k} \\
\epsilon_t &= \left|\frac{a - a_R}{a}\right| \approx \left|\frac{m - \overline{m}}{m}\right| \\
&= \frac{|0.d_{k+1} d_{k+2} \cdots| \times 10^{n-k}}{|0.d_1 d_2 \cdots d_k d_{k+1} d_{k+2} \cdots| \times 10^n} \\
&= \frac{|0.d_{k+1} d_{k+2} \cdots|}{|0.d_1 d_2 \cdots d_k d_{k+1} d_{k+2} \cdots|} \times 10^{-k} \\
&\leq \frac{1}{0.1} \times 10^{-k} = 10^{1-k}
\end{aligned}
$$

Roundoff Error

If $d_{k+1} \geq 5$

$$
\begin{aligned}
E_{tabs} &= |a - a_R| \approx |m - \overline{m}| \times 10^n \\
&= |0.d_1 d_2 \cdots d_k d_{k+1} d_{k+2} \cdots \times 10^n - 0.d_1 d_2 \cdots d_k \times 10^n - 10^{n-k}| \\
&= |0.d_{k+1} d_{k+2} \cdots \times 10^{n-k} - 10^{n-k}| \leq 0.5 \times 10^{n-k} \\
\epsilon_t &= \left|\frac{a - a_R}{a}\right| \approx \left|\frac{m - \overline{m}}{m}\right| \\
&= \frac{|0.d_{k+1} d_{k+2} \cdots \times 10^{n-k} - 10^{n-k}|}{|0.d_1 d_2 \cdots d_k d_{k+1} d_{k+2} \cdots \times 10^n|} \\
&= \frac{|0.d_{k+1} d_{k+2} \cdots - 1|}{|0.d_1 d_2 \cdots d_k d_{k+1} d_{k+2} \cdots|} \times 10^{-k} \\
&\leq \frac{0.5}{0.1} \times 10^{-k} = \frac{1}{2} \times 10^{1-k}
\end{aligned}
$$

Roundoff Error

Definition 4 (Roundoff Error)
Let a_R denote floating point approximation of a obtained by rounding the first k-digits, then the roundoff rule gives the relative error as

$$\epsilon_t = \left|\frac{a - a_R}{a}\right| \approx \left|\frac{m - \overline{m}}{m}\right| \leq \frac{1}{2}10^{1-k}.$$

The right side $u = \frac{1}{2}10^{1-k}$ is called the rounding unit.

If we write $a_R = a(1+\delta)$, we have $\frac{a_R - a}{a} = \delta$. Therefore, $|\delta| \leq u$. This shows that u is an error bound in rounding.

Roundoff

Disadvantages:

- Rounding errors may ruin a computation completely, even a small computation.
- Rounding errors can cause more dangerous problem when millions of arithmetic operations are performed.
- Since digital computer have magnitude and precision limits on their ability to represent numbers, roundoff can cause error when input data are highly sensitive.

Roundoff

Example 5

Obtain the ϵ_t for $\pi = 3.14159265$ for 3 digits, 4 digits and 5 digits.

Solution: $\pi = 3.14159265$

$$a = 0.314159265 \times 10^1$$

For, 3 digits

$$a_R = 0.314 \times 10^1$$

$$\implies \epsilon_t = \frac{|0.314159265 \times 10 - 0.314 \times 10|}{0.314159265 \times 10} = 0.507 \times 10^{-5}$$

For, 4 digits

$$a_R = 0.3142 \times 10^1$$

$$\implies \epsilon_t = \frac{|0.314159265 \times 10 - 0.3142 \times 10|}{0.314159265 \times 10} = 0.1297 \times 10^{-5}$$

Roundoff

Example 6

For, 5 digits

$$a_R = 0.31416 \times 10^1$$

$$\implies \epsilon_t = \frac{|0.314159265 \times 10 - 0.31416 \times 10|}{0.314159265 \times 10} = 0.236 \times 10^{-7}$$

Loss of Significant Digits

Loss of Significant Digits

Although, roundoff errors are inevitable and difficult to control, there are other types of error in computation, that are under our control. A result of calculation has a fewer correct digits than the number from which it was obtained. Loss of significant digits can occur if two number of about the same size, and produces large relative error. For example,

$x = 0.3721478693, y = 0.3720230572, x - y = 0.0001248121$

$$\varepsilon_t = \frac{|x_R - y_R - x + y|}{|x - y|} = 0.04.$$

where 5 significant figures are used. This may occur in simple problems, but it can be avoided in most cases by simple changes in algorithm. To avoid this situations where accuracy can be jeopardized by a subtraction between nearby quantities.

Loss of Significant Digits

Theorem 7 (Loss of Precision Theorem)
If x and y are positive normalized floating point base-2 system such that $x > y > 0$ and

$$2^{-q} \leq 1 - \frac{y}{x} \leq 2^{-p},$$

for some positive integers p and q, then at most q and at least p significant binary digits are lost in subtraction $x - y$.

Proof: Exercise

Loss of Significant Digits

Example 8

For example

$$y = \sqrt{x^2+1} - 1$$

involves subtractive cancellation and loss of significance for small value of x. Consider $x = 10^{-3}$ and five-decimal digit arithmetic. You get $y = 0$. To avoid this, we can rewrite it as

$$y = \sqrt{x^2+1} - 1 \times \frac{\sqrt{x^2+1}+1}{\sqrt{x^2+1}+1} = \frac{x^2}{\sqrt{x^2+1}+1}$$

Now, we get $y = 0.5 \times 10^{-6}$

Loss of Significant Digits

Example 9

In the subtraction $37.593621 - 37.584216$, how many bits of significant digits are lost?

In the subtraction $0.6353 - 0.6311$, how many bits of significant digits are lost?

Loss of Significant Digits

Example 10

How can accurate values of the function

$$f(x) = x - \sin(x)$$

be computed near $x = 0$. Take $x = 10^{-5}$

How can accurate values of the function

$$f(x) = e^{x} - e^{-2x}$$

be computed near $x = 0$. Take $x = 10^{-5}$

Truncation Error

Truncation Error

Truncation errors are those that result from using an approximation of an exact mathematical procedure. For example, the Taylor series for $\sin(x)$ is

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

In practice, it is not possible to use all of the infinite number of terms in the series to compute the sine of angle x. We usually terminate the process after a certain number of terms. The error that results due to such a termination or truncation is called a 'truncation error'. Using Big O notation, we can express this as

$$\sin x = x - \frac{x^3}{6} + O(x^5) \quad (x \to 0).$$

Truncation Error

Usually in evaluating logarithms, exponentials, trigonometric functions, hyperbolic functions etc., an infinite series of the form $f(x) = \sum_{i=0}^{\infty} a_i x^i$ is replaced by a finite sum $P_n(x) = \sum_{i=0}^{n} a_i x^i$. Thus a truncation error of $R_n(x) = \sum_{i=n+1}^{\infty} a_i x^i$ is introduced in the computation.

Truncation Error

Example 11

Consider the evaluation of e^x for the first three terms at $x = 0.2$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \cdots$$

$$e^x \simeq 1 + x + \frac{x^2}{2!}$$

$$e^{0.2} \simeq 1 + 0.2 + \frac{0.04}{2} = 1.22$$

Truncation Error

Example 12

Truncation Error

$$
\begin{aligned}
R_n(x) = \sum_{i=3}^{\infty} \frac{x^i}{i!} &= \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \cdots \\
&= \frac{0.008}{6} + \frac{0.0016}{24} + \cdots \\
&= 0.0013\overline{3} + 0.00006\overline{6} + \cdots \\
&= 0.13\overline{3} \times 10^{-2} + 0.006\overline{6} \times 10^{-2} + \cdots
\end{aligned}
$$

$\therefore$ Truncation Error $\leq 10^{-2}$

Error Propagation

Error Propagation

The relative error ϵ_t seems useless when a is unknown. In this case, we obtain in practice the error bound β_t for $\tilde{a}$, that is, there exists a β_t such that

$$|\epsilon_t| \leq \beta_t.$$

Similarly, for the absolute error, we have an error bound β_{tabs} such that

$$|\epsilon_{tabs}| \leq \beta_{tabs}.$$

It is another important concept in numerical analysis. It deals with how errors at the beginning and in later steps propagate into the computation and affect accuracy, sometimes dangerously.

Error Propagation

Theorem 13
In addition and subtraction, an error bound for the results is given by the sum of the error bounds for the terms.

Theorem 14
In multiplication and division, an error bound for the relative error of the results is given (approximately) by the sum of the bounds for the relative errors of the given numbers.

Algorithm and Stability

Algorithm and Stability

Definition 15 (Algorithm)
An algorithm is a list of unambiguous rules that specify successive steps to solve a problem.

- Numerical methods can be formulated as algorithms
- An algorithm is a step-by-step procedure that expresses a numerical method in a form (a pseudocode) understandable to humans.
- This algorithm is often used to write a program in a programming language that computers can understand so that it can execute the numerical method.

Algorithm and Stability

Stable: An algorithm should be stable, that is, small changes in the initial data should produce only small changes in the final results. If small changes in the initial data produce a large changes in the final results, then the algorithm is unstable.
Numerical instability can be avoided by choosing a better algorithm. Do not confuse between mathematical instability (that is, ill-conditioning) with numerical instability.

Total Numerical Error

Total Numerical Error

- The total numerical error is the summation of the truncation and roundoff errors.
- To minimize the roundoff errors, one has to increase the number of significant figures of the computer.
- As we have discussed, roundoff errors may increase due to subtractive cancellation or due to an increase in the number of computations.
- Truncation error can be reduced by decreasing the step size (will be discussed more detail in Finite Difference Method).

Total Numerical Error

- A decrease in step size can lead to subtractive cancellation or to an increase in computations.
- Truncation errors decrease as the roundoff errors are increased.
- Therefore, decreasing one component of the total error leads to an increase of the other component.
- Thus, finding the appropriate size for a particular computation is a challenging task.

Total Numerical Error

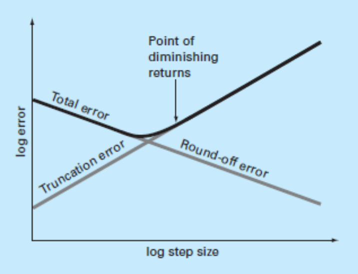

Figure 1: Truncation and Roundoff trade-off

Thanks

Doubts and Suggestions

panch.m@iittp.ac.in

MA633L-Numerical Analysis

Lecture 7 : Chopping, Rounding, Truncation Errors

Panchatcharam Mariappan[1]

[1]Associate Professor
Department of Mathematics and Statistics
IIT Tirupati, Tirupati

January 17, 2025