Numerical Analysis / Numerical Methods

Prof.univ. dr.hab. Viorel Bostan

Lecture 1 (part 3), Spring 2022

Significant digits

Let x_T and x_A be true and, respectively approximated values.

Definition

We say that approximation x_A has m **significant digits** with respect to true value x_T if $|err(x_A)| \leqslant 5$ units in the $(m+1)$-st digit, beginning with the first nonzero digit in x_T.

Significant digits

Let x_T and x_A be true and, respectively approximated values.

Definition

We say that approximation x_A has m **significant digits** with respect to true value x_T if $|err(x_A)| \leqslant 5$ units in the $(m+1)$-st digit, beginning with the first nonzero digit in x_T.

Example

Let

$$
\begin{aligned}
x_T &= e = 2.71828182845904523\ldots, \\
x_A &= \frac{19}{7} = 2.71428571428571428 5\ldots, \\
err(x_A) &\approx 0.003996\ldots
\end{aligned}
$$

Therefore, there are 3 significant digits in this approximation.

Significant digits

Another way to look at significant digits (also called significant figures) regardless whether we have an approximation or not is each of the digits of a number that are used to express it to the required degree of accuracy, starting from the first non-zero digit.

There are several rules:

- All non-zero numbers are significant.
  The number 37.9 has 3 significant digits.
- Zeros between two non-zero digits are significant.
  The number 4001.7 has 5 significant digits,
  while the number 2005 has 4 significant digits.
- Leading zeros are not significant.
  The number 0.89 has only 2 significant digits, and 0.00017 also has 2 significant digits.
- Trailing zeros to the right of the decimal are significant.
  There are 4 significant digits in 92.00 and 5 significant digits in 3.0000.

Loss of significance errors

This can be considered a source of error or a consequence of the finiteness of calculator and computer arithmetic.

Loss of significance errors

This can be considered a source of error or a consequence of the finiteness of calculator and computer arithmetic.

Example 1. Define

$$f(x) = x\left(\sqrt{x+1} - \sqrt{x}\right)$$

and consider evaluating it on a 6-digit decimal calculator which uses rounded arithmetic.

This can be considered a source of error or a consequence of the finiteness of calculator and computer arithmetic.

Example 1. Define

$$f(x) = x\left(\sqrt{x+1} - \sqrt{x}\right)$$

and consider evaluating it on a 6-digit decimal calculator which uses rounded arithmetic.

x	Computed $f(x)$	True $f(x)$	Error
1	0.4142210	0.414214	$7.0000e-006$
10	1.54340	1.54347	$-7.0000e-005$
100	4.99000	4.98756	0.0024
1000	15.8000	15.8074	-0.0074
10000	50.0000	49.9988	0.0012
100000	100.000	158.113	-58.1130

In order to localize the error, consider the case $x = 100$.

Loss of significance errors. Example 1

In order to localize the error, consider the case $x = 100$.

The calculator with 6 decimal digits will provide us with the following values

$$\sqrt{100} = 10, \quad \sqrt{101} = 10.0499.$$

Loss of significance errors. Example 1

In order to localize the error, consider the case $x = 100$.

The calculator with 6 decimal digits will provide us with the following values

$$\sqrt{100} = 10, \quad \sqrt{101} = 10.0499.$$

Then,

$$\sqrt{x+1} - \sqrt{x} = \sqrt{101} - \sqrt{100} = 0.0499000,$$

while the exact value is 0.0498756.

Loss of significance errors. Example 1

In order to localize the error, consider the case $x = 100$.

The calculator with 6 decimal digits will provide us with the following values

$$\sqrt{100} = 10, \quad \sqrt{101} = 10.0499.$$

Then,

$$\sqrt{x+1} - \sqrt{x} = \sqrt{101} - \sqrt{100} = 0.0499000,$$

while the exact value is 0.0498756.

Three significant digits in $\sqrt{x+1} = \sqrt{101}$ have been lost from $\sqrt{x} = \sqrt{100}$.

Loss of significance errors. Example 1

In order to localize the error, consider the case $x = 100$.

The calculator with 6 decimal digits will provide us with the following values

$$\sqrt{100} = 10, \quad \sqrt{101} = 10.0499.$$

Then,

$$\sqrt{x+1} - \sqrt{x} = \sqrt{101} - \sqrt{100} = 0.0499000,$$

while the exact value is 0.0498756.

Three significant digits in $\sqrt{x+1} = \sqrt{101}$ have been lost from $\sqrt{x} = \sqrt{100}$.

The loss of precision is due to the form of the function $f(x)$ and the finiteness of the precision of the 6 digit calculator.

Loss of significance errors. Example 2

In this particular case, we can avoid the loss of precision by rewriting the function as follows:

In this particular case, we can avoid the loss of precision by rewriting the function as follows:

$$\begin{aligned} f(x) &= x\frac{\sqrt{x+1}+\sqrt{x}}{\sqrt{x+1}+\sqrt{x}} \cdot \frac{\sqrt{x+1}-\sqrt{x}}{1} \\ &= \frac{x}{\sqrt{x+1}+\sqrt{x}}. \end{aligned}$$

Thus we will avoid the subtraction on near quantities.

Doing so gives us

$$f(100) = 4.98756,$$

a value with 6 significant digits.

Example 2. Define

$$g(x) = \frac{1 - \cos x}{x^2}$$

and consider evaluating it on a 10-digit decimal calculator which uses rounded arithmetic.

Example 2. Define

$$g(x) = \frac{1 - \cos x}{x^2}$$

and consider evaluating it on a 10-digit decimal calculator which uses rounded arithmetic.

x	Computed $f(x)$	True $f(x)$	Error
0.1	0.4995834700	0.4995834722	$-2.2000e-009$
0.01	0.4999960000	0.4999958333	$1.6670e-007$
0.001	0.5000000000	0.4999999583	$4.1700e-008$
0.0001	0.5000000000	0.4999999996	$4.0000e-010$
0.00001	0.0	0.5000000000	0.5

Loss of significance errors. Example 2

Consider one case, that of $x = 0.001$.

Loss of significance errors. Example 2

Consider one case, that of $x = 0.001$.

Then on the calculator:

$$
\begin{aligned}
\cos(0.001) &= 0.9999994999 \\
1 - \cos(0.001) &= 5.001 \times 10^{-7} \\
\frac{1 - \cos(0.001)}{(0.001)^2} &= 0.5001000000
\end{aligned}
$$

Loss of significance errors. Example 2

Consider one case, that of $x = 0.001$.

Then on the calculator:

$$\begin{aligned} \cos(0.001) &= 0.9999994999 \\ 1 - \cos(0.001) &= 5.001 \times 10^{-7} \\ \frac{1 - \cos(0.001)}{(0.001)^2} &= 0.5001000000 \end{aligned}$$

The true answer is

$$f(0.001) = 0.4999999583$$

Loss of significance errors. Example 2

Consider one case, that of $x = 0.001$.

Then on the calculator:

$$\begin{aligned}
\cos(0.001) &= 0.9999994999 \\
1 - \cos(0.001) &= 5.001 \times 10^{-7} \\
\frac{1 - \cos(0.001)}{(0.001)^2} &= 0.5001000000
\end{aligned}$$

The true answer is

$$f(0.001) = 0.4999999583$$

The relative error in our answer is

$$\frac{0.4999999583 - 0.5001}{0.4999999583} = \frac{-0.0001000417}{0.4999999583} = -0.0002$$

Consider one case, that of $x = 0.001$.

Then on the calculator:

$$\begin{aligned} \cos(0.001) &= 0.9999994999 \\ 1 - \cos(0.001) &= 5.001 \times 10^{-7} \\ \frac{1 - \cos(0.001)}{(0.001)^2} &= 0.5001000000 \end{aligned}$$

The true answer is

$$f(0.001) = 0.4999999583$$

The relative error in our answer is

$$\frac{0.4999999583 - 0.5001}{0.4999999583} = \frac{-0.0001000417}{0.4999999583} = -0.0002$$

There are 3 significant digits in the answer.

How can such a straightforward and short calculation lead to such a large error (relative to the accuracy of the calculator)?

Loss of significance errors. Example 2

When two numbers are nearly equal and we subtract them, then we suffer a “loss of significance error” in the calculation.

Loss of significance errors. Example 2

When two numbers are nearly equal and we subtract them, then we suffer a “loss of significance error” in the calculation.

In some cases, these can be quite subtle and difficult to detect.

Loss of significance errors. Example 2

When two numbers are nearly equal and we subtract them, then we suffer a “loss of significance error” in the calculation.

In some cases, these can be quite subtle and difficult to detect.

And even after they are detected, they may be difficult to fix.

Loss of significance errors. Example 2

When two numbers are nearly equal and we subtract them, then we suffer a "loss of significance error" in the calculation.

In some cases, these can be quite subtle and difficult to detect.

And even after they are detected, they may be difficult to fix.

The last example, fortunately, can be fixed in a number of ways. Easiest is to use a trigonometric identity:

$$\cos(x) = 1 - 2\sin^2(x/2),$$

$$f(x) = \frac{1-\cos x}{x^2} = \frac{2\sin^2(x/2)}{x^2} = \frac{1}{2}\left(\frac{\sin(x/2)}{x/2}\right)^2.$$

This latter formula, with $x = 0.001$, yields a computed value of 0.4999999584, nearly the true answer. We could also have used a Taylor polynomial for $\cos(x)$ around $x = 0$ to obtain a better approximation to $f(x)$ for small values of x.

Example 3. Evaluate e^{-5} using a Taylor polynomial approximation:

$$e^{-5} = 1 + \frac{(-5)}{1!} + \frac{(-5)^2}{2!} + \frac{(-5)^3}{3!} + \frac{(-5)^4}{4!} + \frac{(-5)^5}{5!} + \frac{(-5)^6}{6!} \dots$$

Example 3. Evaluate e^{-5} using a Taylor polynomial approximation:

$$e^{-5} = 1 + \frac{(-5)}{1!} + \frac{(-5)^2}{2!} + \frac{(-5)^3}{3!} + \frac{(-5)^4}{4!} + \frac{(-5)^5}{5!} + \frac{(-5)^6}{6!} \dots$$

With $n = 25$, the error is

$$\left| \frac{(-5)^{-26}}{26!} e^c \right| \leq 10^{-8}.$$

Example 3. Evaluate e^{-5} using a Taylor polynomial approximation:

$$e^{-5} = 1 + \frac{(-5)}{1!} + \frac{(-5)^2}{2!} + \frac{(-5)^3}{3!} + \frac{(-5)^4}{4!} + \frac{(-5)^5}{5!} + \frac{(-5)^6}{6!} \dots$$

With $n = 25$, the error is

$$\left| \frac{(-5)^{-26}}{26!} e^c \right| \leq 10^{-8}.$$

Imagine calculating this polynomial using a computer with 4 digit decimal arithmetic and rounding.

Loss of significance errors. Example 3

Example 3. Evaluate e^{-5} using a Taylor polynomial approximation:

$$e^{-5} = 1 + \frac{(-5)}{1!} + \frac{(-5)^2}{2!} + \frac{(-5)^3}{3!} + \frac{(-5)^4}{4!} + \frac{(-5)^5}{5!} + \frac{(-5)^6}{6!} \dots$$

With $n = 25$, the error is

$$\left| \frac{(-5)^{-26}}{26!} e^c \right| \leq 10^{-8}.$$

Imagine calculating this polynomial using a computer with 4 digit decimal arithmetic and rounding.

To make the point about cancellation more strongly, imagine that each of the terms in the above polynomial is calculated exactly and then rounded to the arithmetic of the computer. We add the terms exactly and then we round to four digits.

Loss of significance errors. Example 3

Degree	Term	Sum	Degree	Term	Sum
0	1.000	1.000	13	-0.1960	-0.04230
1	-5.000	-4.000	14	$0.7001e-1$	0.02771
2	12.50	8.500	15	$-0.2334e-1$	0.004370
3	-20.83	-12.33	16	$0.7293e-2$	0.01166
4	26.04	13.71	17	$-0.2145e-2$	0.009518
5	-26.04	-12.33	18	$0.5958e-3$	0.01011
6	21.70	9.370	19	$-0.1568e-3$	0.009957
7	-15.50	-6.130	20	$0.3920e-4$	0.009996
8	9.688	3.558	21	$-0.9333e-5$	0.009987
9	-5.382	-1.824	22	$0.2121e-5$	0.009989
10	2.691	0.8670	23	$-0.4611e-6$	0.009989
11	-1.223	-0.3560	24	$0.9670e-7$	0.009989
12	0.5097	0.1537	25	$-0.1921e-7$	0.009989

Loss of significance errors. Example 3

Degree	Term	Sum	Degree	Term	Sum
0	1.000	1.000	13	-0.1960	-0.04230
1	-5.000	-4.000	14	$0.7001e-1$	0.02771
2	12.50	8.500	15	$-0.2334e-1$	0.004370
3	-20.83	-12.33	16	$0.7293e-2$	0.01166
4	26.04	13.71	17	$-0.2145e-2$	0.009518
5	-26.04	-12.33	18	$0.5958e-3$	0.01011
6	21.70	9.370	19	$-0.1568e-3$	0.009957
7	-15.50	-6.130	20	$0.3920e-4$	0.009996
8	9.688	3.558	21	$-0.9333e-5$	0.009987
9	-5.382	-1.824	22	$0.2121e-5$	0.009989
10	2.691	0.8670	23	$-0.4611e-6$	0.009989
11	-1.223	-0.3560	24	$0.9670e-7$	0.009989
12	0.5097	0.1537	25	$-0.1921e-7$	0.009989

True answer is 0.006738!

Loss of significance errors. Example 3

Look at the numbers being added and their accuracy, ex. 3rd term:

$$\frac{(-5)^3}{3!} = -\frac{125}{6} = -20.83$$

in the 4 digit decimal calculation, with an error of magnitude 0.00333.

Loss of significance errors. Example 3

Look at the numbers being added and their accuracy, ex. 3rd term:

$$\frac{(-5)^3}{3!} = -\frac{125}{6} = -20.83$$

in the 4 digit decimal calculation, with an error of magnitude 0.00333.

Note that this error in an intermediate step is of same magnitude as the true answer 0.006738 being sought.

Loss of significance errors. Example 3

Look at the numbers being added and their accuracy, ex. 3rd term:

$$\frac{(-5)^3}{3!} = -\frac{125}{6} = -20.83$$

in the 4 digit decimal calculation, with an error of magnitude 0.00333.

Note that this error in an intermediate step is of same magnitude as the true answer 0.006738 being sought.

Other similar errors are present in calculating other coefficients, and thus they cause a major error in the final answer being calculated.

Loss of significance errors. Example 3

Look at the numbers being added and their accuracy, ex. 3rd term:

$$\frac{(-5)^3}{3!} = -\frac{125}{6} = -20.83$$

in the 4 digit decimal calculation, with an error of magnitude 0.00333.

Note that this error in an intermediate step is of same magnitude as the true answer 0.006738 being sought.

Other similar errors are present in calculating other coefficients, and thus they cause a major error in the final answer being calculated.

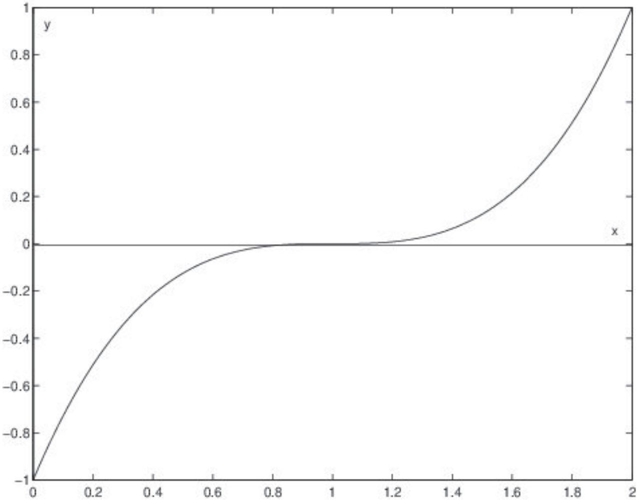
General principle

Whenever a sum is being formed in which the final answer is much smaller than some of the terms being combined, then a loss of significance error is occurring.

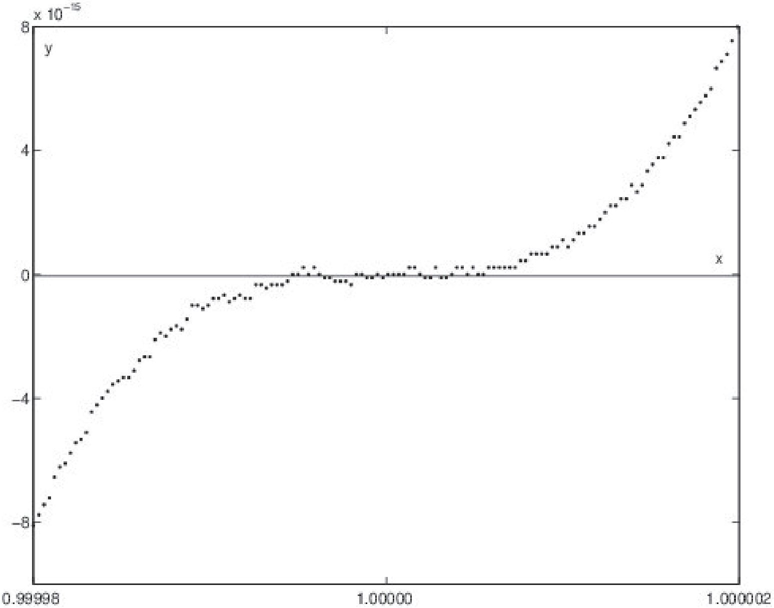
Noise in function evaluation

Consider plotting the function

$$f(x) = (x-1)^3 = x^3 - 3x^2 + 3x - 1 = -1 + x(3 + x(-3 + x)).$$



Noise in function evaluation



Noise in function evaluation

Whenever a function $f(x)$ is evaluated, there are arithmetic operations carried out which involve rounding or chopping errors.

Noise in function evaluation

Whenever a function $f(x)$ is evaluated, there are arithmetic operations carried out which involve rounding or chopping errors.

This means that what the computer eventually returns as an answer contains noise.

Noise in function evaluation

Whenever a function $f(x)$ is evaluated, there are arithmetic operations carried out which involve rounding or chopping errors.

This means that what the computer eventually returns as an answer contains noise.

This noise is generally “random” and small.

Noise in function evaluation

Whenever a function $f(x)$ is evaluated, there are arithmetic operations carried out which involve rounding or chopping errors.

This means that what the computer eventually returns as an answer contains noise.

This noise is generally "random" and small.

But it can affect the accuracy of other calculations which depend on $f(x)$.

Underflow errors

Consider evaluating the function

$$f(x) = x^{10}$$

for values of x close to 0.

Underflow errors

Consider evaluating the function

$$f(x) = x^{10}$$

for values of x close to 0.

When using IEEE single precision arithmetic, the smallest nonzero positive number expressible (stored exactly) in normalized floating-point format is

$$m = 2^{-126} \approx 1.18 \times 10^{-38}.$$

Underflow errors

Consider evaluating the function

$$f(x) = x^{10}$$

for values of x close to 0.

When using IEEE single precision arithmetic, the smallest nonzero positive number expressible (stored exactly) in normalized floating-point format is

$$m = 2^{-126} \approx 1.18 \times 10^{-38}.$$

Thus, $f(x)$ will be stored as zero if

$$\begin{aligned} x^{10} &< m, \\ |x| &< m^{\frac{1}{10}}, \\ |x| &< 1.61 \times 10^{-4}, \\ -0.000161 < x &< 0.000161. \end{aligned}$$

Storing a nonzero number as a zero leads to **underflow error**.

Overflow errors

Attempts to use numbers that are too large for the floating-point format will lead to **overflow errors**.

Overflow errors

Attempts to use numbers that are too large for the floating-point format will lead to **overflow errors**.

These are generally fatal errors on most computers. With the IEEE floating-point format, overflow errors can be carried along as having a value of $\pm\infty$ or NaN, depending on the context.

Overflow errors

Attempts to use numbers that are too large for the floating-point format will lead to **overflow errors**.

These are generally fatal errors on most computers. With the IEEE floating-point format, overflow errors can be carried along as having a value of $\pm\infty$ or NaN, depending on the context.

Usually an overflow error is an indication of a more significant problem and the user needs to be aware of such errors.

Overflow errors

Attempts to use numbers that are too large for the floating-point format will lead to **overflow errors**.

These are generally fatal errors on most computers. With the IEEE floating-point format, overflow errors can be carried along as having a value of $\pm\infty$ or NaN, depending on the context.

Usually an overflow error is an indication of a more significant problem and the user needs to be aware of such errors.

When using IEEE single precision arithmetic, the largest nonzero positive number expressible in normalized floating point format is

$$m = 2^{128}\left(1 - 2^{-24}\right) = 3.40 \times 10^{38}$$

Overflow errors

Attempts to use numbers that are too large for the floating-point format will lead to **overflow errors**.

These are generally fatal errors on most computers. With the IEEE floating-point format, overflow errors can be carried along as having a value of $\pm\infty$ or NaN, depending on the context.

Usually an overflow error is an indication of a more significant problem and the user needs to be aware of such errors.

When using IEEE single precision arithmetic, the largest nonzero positive number expressible in normalized floating point format is

$$m = 2^{128}\left(1 - 2^{-24}\right) = 3.40 \times 10^{38}$$

Thus, $f(x)$ will overflow if

$$\begin{aligned} x^{10} &> m, \\ |x| &> m^{\frac{1}{10}}, \\ |x| &> 7131.6 \end{aligned}$$

Propagation of arithmetic operations errors

Let ω denote arithmetic operation such as $+, -, *$,or $/$.

Propagation of arithmetic operations errors

Let ω denote arithmetic operation such as $+, -, *$,or $/$.

Let ω^* denote the same arithmetic operation as it is actually carried out in the computer, including rounding or chopping error.

Propagation of arithmetic operations errors

Let ω denote arithmetic operation such as $+$, $-$, $*$,or $/$.

Let ω^* denote the same arithmetic operation as it is actually carried out in the computer, including rounding or chopping error.

Let $x_A \approx x_T$ and $y_A \approx y_T$.

Propagation of arithmetic operations errors

Let ω denote arithmetic operation such as $+, -, *,$or $/$.

Let ω^* denote the same arithmetic operation as it is actually carried out in the computer, including rounding or chopping error.

Let $x_A \approx x_T$ and $y_A \approx y_T$.

We want to obtain $x_T \, \omega \, y_T$, but we actually obtain $x_A \, \omega^* \, y_A$.

Propagation of arithmetic operations errors

Let ω denote arithmetic operation such as $+$, $-$, $*$,or $/$.

Let ω^* denote the same arithmetic operation as it is actually carried out in the computer, including rounding or chopping error.

Let $x_A \approx x_T$ and $y_A \approx y_T$.

We want to obtain $x_T \, \omega \, y_T$, but we actually obtain $x_A \, \omega^* \, y_A$.

The error in $x_A \, \omega^* \, y_A$ is given by $x_T \, \omega \, y_T - x_A \, \omega^* \, y_A$.

Propagation of arithmetic operations errors

Let ω denote arithmetic operation such as $+$, $-$, $*$,or $/$.

Let ω^* denote the same arithmetic operation as it is actually carried out in the computer, including rounding or chopping error.

Let $x_A \approx x_T$ and $y_A \approx y_T$.

We want to obtain $x_T \, \omega \, y_T$, but we actually obtain $x_A \, \omega^* \, y_A$.

The error in $x_A \, \omega^* \, y_A$ is given by $x_T \, \omega \, y_T - x_A \, \omega^* \, y_A$.
The error in $x_A \, \omega^* \, y_A$ can be rewritten as:

$$x_T \, \omega \, y_T - x_A \, \omega^* \, y_A = \left[x_T \, \omega \, y_T - x_A \, \omega \, y_A\right] + \left[x_A \, \omega \, y_A - x_A \, \omega^* \, y_A\right]$$

The last term is the error introduced by the inexactness of the machine arithmetic, since it can be shown that

$$\mathrm{Rel}(x_A \, \omega^* \, y_A) = -\varepsilon.$$

With rounded binary arithmetic having n digits in the mantissa,

$$-2^{-n} \leq \varepsilon \leq 2^{-n}.$$

Propagation of arithmetic operations errors

With rounded binary arithmetic having n digits in the mantissa,

$$-2^{-n} \le \varepsilon \le 2^{-n}.$$

Coming back to error formula we have

$$x_T\,\omega\,y_T - x_A\,\omega^*\,y_A = \left[x_T\,\omega\,y_T - x_A\,\omega\,y_A\right] + \underbrace{\left[x_A\,\omega\,y_A - x_A\,\omega^*\,y_A\right]}_{\text{Relative error is } -\varepsilon}.$$

Propagation of arithmetic operations errors

With rounded binary arithmetic having n digits in the mantissa,

$$-2^{-n} \leq \varepsilon \leq 2^{-n}.$$

Coming back to error formula we have

$$x_T\,\omega\,y_T - x_A\,\omega^*\,y_A = \left[x_T\,\omega\,y_T - x_A\,\omega\,y_A\right] + \underbrace{\left[x_A\,\omega\,y_A - x_A\,\omega^*\,y_A\right]}_{\text{Relative error is } -\varepsilon}.$$

The first term from the right side

$$x_T\,\omega\,y_T - x_A\,\omega\,y_A$$

is called the **propagated error**.

With rounded binary arithmetic having n digits in the mantissa,

$$-2^{-n} \leq \varepsilon \leq 2^{-n}.$$

Coming back to error formula we have

$$x_T \, \omega \, y_T - x_A \, \omega^* \, y_A = \left[x_T \, \omega \, y_T - x_A \, \omega \, y_A\right] + \underbrace{\left[x_A \, \omega \, y_A - x_A \, \omega^* \, y_A\right]}_{\text{Relative error is } -\varepsilon}.$$

The first term from the right side

$$x_T \, \omega \, y_T - x_A \, \omega \, y_A$$

is called the **propagated error**.

In what follows we examine it for particular cases.

With rounded binary arithmetic having n digits in the mantissa,

$$-2^{-n} \leq \varepsilon \leq 2^{-n}.$$

Coming back to error formula we have

$$x_T \,\omega\, y_T - x_A \,\omega^*\, y_A = \left[x_T \,\omega\, y_T - x_A \,\omega\, y_A\right] + \underbrace{\left[x_A \,\omega\, y_A - x_A \,\omega^*\, y_A\right]}_{\text{Relative error is } -\varepsilon}.$$

The first term from the right side

$$x_T \,\omega\, y_T - x_A \,\omega\, y_A$$

is called the **propagated error**.

In what follows we examine it for particular cases.

Let $\omega = *$ (i.e. multiplication). Write

$$x_T = x_A + \xi, \qquad y_T = y_A + \eta,$$

where ξ and η are the approximation errors is x_A and y_A.

Propagation of arithmetic operations errors

Then for the relative error in $x_A\, y_A$

$$\begin{aligned}
\mathrm{Rel}(x_A * y_A) &= \frac{x_T * y_T - x_A * y_A}{x_T * y_T} \\
&= \frac{x_T * y_T - (x_T - \xi) * (y_T - \eta)}{x_T * y_T} \\
&= \frac{x_T \eta + y_T \xi - \xi \eta}{x_T * y_T} \\
&= \frac{\xi}{x_T} + \frac{\eta}{y_T} - \frac{\xi}{x_T} \cdot \frac{\eta}{y_T} \\
&= \mathrm{Rel}(x_A) + \mathrm{Rel}(y_A) - \mathrm{Rel}(x_A) \cdot \mathrm{Rel}(y_A).
\end{aligned}$$

Then for the relative error in $x_A\, y_A$

$$\begin{aligned}
\mathrm{Rel}(x_A * y_A) &= \frac{x_T * y_T - x_A * y_A}{x_T * y_T} \\
&= \frac{x_T * y_T - (x_T - \xi) * (y_T - \eta)}{x_T * y_T} \\
&= \frac{x_T\eta + y_T\xi - \xi\eta}{x_T * y_T} \\
&= \frac{\xi}{x_T} + \frac{\eta}{y_T} - \frac{\xi}{x_T} \cdot \frac{\eta}{y_T} \\
&= \mathrm{Rel}(x_A) + \mathrm{Rel}(y_A) - \mathrm{Rel}(x_A) \cdot \mathrm{Rel}(y_A).
\end{aligned}$$

Usually we have $|\mathrm{Rel}(x_A)| \ll 1, \quad |\mathrm{Rel}(y_A)| \ll 1$. Therefore, we can skip the last term $\mathrm{Rel}(x_A) \cdot \mathrm{Rel}(\, y_A)$, since it is much smaller compared with previous two.

$$\begin{aligned}
\mathrm{Rel}(x_A y_A) &= Rel(x_A) + Rel(y_A) - \mathrm{Rel}(x_A) \cdot \mathrm{Rel}(y_A) \\
&\approx \mathrm{Rel}(x_A) + \mathrm{Rel}(y_A).
\end{aligned}$$

Thus, in multiplication small relative errors in the arguments x_A and y_A lead to a small relative error (the sum of relative errors from factors) in the product $x_A * y_A$.

Thus, in multiplication small relative errors in the arguments x_A and y_A lead to a small relative error (the sum of relative errors from factors) in the product $x_A * y_A$.

Also, note that there will be some cancellation, if these relative errors are of opposite sign.

Propagation of arithmetic operations errors

Thus, in multiplication small relative errors in the arguments x_A and y_A lead to a small relative error (the sum of relative errors from factors) in the product $x_A * y_A$.

Also, note that there will be some cancellation, if these relative errors are of opposite sign.

There is a similar result for division:

$$\mathrm{Rel}(x_A\, y_A) \approx \mathrm{Rel}(x_A) - \mathrm{Rel}(\, y_A)$$

provided $|\mathrm{Rel}(y_A)| \ll 1$.

Propagation of arithmetic operations errors

Thus, in multiplication small relative errors in the arguments x_A and y_A lead to a small relative error (the sum of relative errors from factors) in the product $x_A * y_A$.

Also, note that there will be some cancellation, if these relative errors are of opposite sign.

There is a similar result for division:

$$\mathrm{Rel}(x_A\, y_A) \approx \mathrm{Rel}(x_A) - \mathrm{Rel}(\, y_A)$$

provided $|\mathrm{Rel}(y_A)| \ll 1$.

For ω equal to $-$ or $+$, we have

$$[x_T \pm y_T] - [x_A \pm y_A] = [x_T - x_A] \pm [y_T - y_A].$$

Thus, in multiplication small relative errors in the arguments x_A and y_A lead to a small relative error (the sum of relative errors from factors) in the product $x_A * y_A$.

Also, note that there will be some cancellation, if these relative errors are of opposite sign.

There is a similar result for division:

$$\mathrm{Rel}(x_A\, y_A) \approx \mathrm{Rel}(x_A) - \mathrm{Rel}(\, y_A)$$

provided $|\mathrm{Rel}(y_A)| \ll 1$.

For ω equal to $-$ or $+$, we have

$$[x_T \pm y_T] - [x_A \pm y_A] = [x_T - x_A] \pm [y_T - y_A].$$

Thus, **the error in a sum is the sum of the errors in the original arguments, and similarly for subtraction.**

Propagation of arithmetic operations errors

Thus, in multiplication small relative errors in the arguments x_A and y_A lead to a small relative error (the sum of relative errors from factors) in the product $x_A * y_A$.

Also, note that there will be some cancellation, if these relative errors are of opposite sign.

There is a similar result for division:

$$\mathrm{Rel}(x_A\, y_A) \approx \mathrm{Rel}(x_A) - \mathrm{Rel}(\, y_A)$$

provided $|\mathrm{Rel}(y_A)| \ll 1$.

For ω equal to $-$ or $+$, we have

$$[x_T \pm y_T] - [x_A \pm y_A] = [x_T - x_A] \pm [y_T - y_A].$$

Thus, **the error in a sum is the sum of the errors in the original arguments, and similarly for subtraction.**

However, there is a more subtle error occurring here.

Errors in function evaluations

Suppose we are evaluating a function $f(x)$ in the machine.

Errors in function evaluations

Suppose we are evaluating a function $f(x)$ in the machine.

Then, the result is generally not $f(x)$, but rather an approximate of it, which we denote by $\widetilde{f}(x)$.

Errors in function evaluations

Suppose we are evaluating a function $f(x)$ in the machine.

Then, the result is generally not $f(x)$, but rather an approximate of it, which we denote by $\widetilde{f}(x)$.

Now, suppose that we have a number $x_A \approx x_T$.

Errors in function evaluations

Suppose we are evaluating a function $f(x)$ in the machine.

Then, the result is generally not $f(x)$, but rather an approximate of it, which we denote by $\widetilde{f}(x)$.

Now, suppose that we have a number $x_A \approx x_T$.

We want to calculate $f(x_T)$, but instead we evaluate $\widetilde{f}(x_A)$.

Errors in function evaluations

Suppose we are evaluating a function $f(x)$ in the machine.

Then, the result is generally not $f(x)$, but rather an approximate of it, which we denote by $\widetilde{f}(x)$.

Now, suppose that we have a number $x_A \approx x_T$.

We want to calculate $f(x_T)$, but instead we evaluate $\widetilde{f}(x_A)$.

What can we say about the error in this latter computed quantity?

Rewrite the error

$$f(x_T) - \widetilde{f}(x_A) = \left[f(x_T) - f(x_A)\right] + \left[f(x_A) - \widetilde{f}(x_A)\right].$$

Suppose we are evaluating a function $f(x)$ in the machine.

Then, the result is generally not $f(x)$, but rather an approximate of it, which we denote by $\widetilde{f}(x)$.

Now, suppose that we have a number $x_A \approx x_T$.

We want to calculate $f(x_T)$, but instead we evaluate $\widetilde{f}(x_A)$.

What can we say about the error in this latter computed quantity?

Rewrite the error

$$f(x_T) - \widetilde{f}(x_A) = \left[f(x_T) - f(x_A)\right] + \left[f(x_A) - \widetilde{f}(x_A)\right].$$

The quantity $f(x_A) - \widetilde{f}(x_A)$ is the **noise** in the evaluation of $f(x_A)$ in the computer.

Errors in function evaluations

Suppose we are evaluating a function $f(x)$ in the machine.

Then, the result is generally not $f(x)$, but rather an approximate of it, which we denote by $\widetilde{f}(x)$.

Now, suppose that we have a number $x_A \approx x_T$.

We want to calculate $f(x_T)$, but instead we evaluate $\widetilde{f}(x_A)$.

What can we say about the error in this latter computed quantity?

Rewrite the error

$$f(x_T) - \widetilde{f}(x_A) = \left[f(x_T) - f(x_A)\right] + \left[f(x_A) - \widetilde{f}(x_A)\right].$$

The quantity $f(x_A) - \widetilde{f}(x_A)$ is the **noise** in the evaluation of $f(x_A)$ in the computer.

The quantity $f(x_T) - f(x_A)$ is called the **propagated error**. It is the error that results from using perfect arithmetic in the evaluation of the function.

If the function $f(x)$ is differentiable, then we can use the **Mean-value Theorem** from calculus to write

$$f(x_T) - f(x_A) = f'(\xi)(x_T - x_A)$$

for some ξ between x_T andx_A.

If the function $f(x)$ is differentiable, then we can use the **Mean-value Theorem** from calculus to write

$$f(x_T) - f(x_A) = f'(\xi)(x_T - x_A)$$

for some ξ between x_T andx_A.

Since usually x_T and x_A are close together, we can say ξ is close to either of them, and

$$\begin{aligned} f(x_T) - f(x_A) &= f'(\xi)(x_T - x_A) \\ &\approx f'(x_T)(x_T - x_A) \\ &\approx f'(x_A)(x_T - x_A). \end{aligned}$$

If the function $f(x)$ is differentiable, then we can use the **Mean-value Theorem** from calculus to write

$$f(x_T) - f(x_A) = f'(\xi)(x_T - x_A)$$

for some ξ between x_T andx_A.

Since usually x_T and x_A are close together, we can say ξ is close to either of them, and

$$\begin{aligned} f(x_T) - f(x_A) &= f'(\xi)(x_T - x_A) \\ &\approx f'(x_T)(x_T - x_A) \\ &\approx f'(x_A)(x_T - x_A). \end{aligned}$$

This last approximation can be used in practice to estimate the propagated error once function f and its derivative are known.

Example. Define $f(x) = b^x$, where b is a positive real number. Then, last formula yields

$$b^{x_T} - b^{x_A} \approx (\ln b) b^{x_T} (x_T - x_A).$$

Example. Define $f(x) = b^x$, where b is a positive real number. Then, last formula yields

$$b^{x_T} - b^{x_A} \approx (\ln b) b^{x_T} (x_T - x_A).$$

Therefore,

$$\begin{aligned} \operatorname{Rel}\left(b^{x_A}\right) &\approx \frac{(\ln b) b^{x_T} (x_T - x_A)}{b^{x_T}} \\ &= \frac{(\ln b)(x_T - x_A) x_T}{x_T} \\ &= x_T \ln b \cdot \operatorname{Rel}(x_A) \\ &= K \cdot \operatorname{Rel}(x_A). \end{aligned}$$

Errors in function evaluations. Example

Example. Define $f(x) = b^x$, where b is a positive real number. Then, last formula yields

$$b^{x_T} - b^{x_A} \approx (\ln b) b^{x_T} (x_T - x_A).$$

Therefore,

$$\begin{aligned} \mathrm{Rel}\left(b^{x_A}\right) &\approx \frac{(\ln b) b^{x_T} (x_T - x_A)}{b^{x_T}} \\ &= \frac{(\ln b)(x_T - x_A) x_T}{x_T} \\ &= x_T \ln b \cdot \mathrm{Rel}(x_A) \\ &= K \cdot \mathrm{Rel}(x_A). \end{aligned}$$

Note that if $K = 10^4$ and $\mathrm{Rel}(x_A) = 10^{-7}$, then $\mathrm{Rel}(b^{x_A}) \approx 10^{-3}$.

Example. Define $f(x) = b^x$, where b is a positive real number. Then, last formula yields

$$b^{x_T} - b^{x_A} \approx (\ln b) b^{x_T} (x_T - x_A).$$

Therefore,

$$\begin{aligned}
\mathrm{Rel}\left(b^{x_A}\right) &\approx \frac{(\ln b) b^{x_T} (x_T - x_A)}{b^{x_T}} \\
&= \frac{(\ln b)(x_T - x_A) x_T}{x_T} \\
&= x_T \ln b \cdot \mathrm{Rel}(x_A) \\
&= K \cdot \mathrm{Rel}(x_A).
\end{aligned}$$

Note that if $K = 10^4$ and $\mathrm{Rel}(x_A) = 10^{-7}$, then $\mathrm{Rel}(b^{x_A}) \approx 10^{-3}$.

This is a large decrease in accuracy; and it is independent of how we actually calculate b^x.

Errors in function evaluations. Example

Example. Define $f(x) = b^x$, where b is a positive real number. Then, last formula yields

$$b^{x_T} - b^{x_A} \approx (\ln b) b^{x_T} (x_T - x_A).$$

Therefore,

$$\begin{aligned}
\mathrm{Rel}\,(b^{x_A}) &\approx \frac{(\ln b) b^{x_T} (x_T - x_A)}{b^{x_T}} \\
&= \frac{(\ln b)(x_T - x_A) x_T}{x_T} \\
&= x_T \ln b \cdot \mathrm{Rel}(x_A) \\
&= K \cdot \mathrm{Rel}(x_A).
\end{aligned}$$

Note that if $K = 10^4$ and $\mathrm{Rel}(x_A) = 10^{-7}$, then $\mathrm{Rel}(b^{x_A}) \approx 10^{-3}$.

This is a large decrease in accuracy; and it is independent of how we actually calculate b^x.

The number K is called a **condition number** for the computation.

Summation

Let S be a sum with a relatively large number of terms

$$S = a_1 + a_2 + \ldots a_n, \tag{1}$$

where $\{a_j\}_{j=1}^{n}$ are floating point numbers.

Summation

Let S be a sum with a relatively large number of terms

$$S = a_1 + a_2 + \dots a_n, \tag{1}$$

where $\{a_j\}_{j=1}^{n}$ are floating point numbers. The summation process in (1) consists of $n-1$ consecutive additions:

$$S = (((\dots(a_1 + a_2) + a_3) + \dots + a_{n-1}) + a_n.$$

Let S be a sum with a relatively large number of terms

$$S = a_1 + a_2 + \dots a_n, \tag{1}$$

where $\{a_j\}_{j=1}^{n}$ are floating point numbers. The summation process in (1) consists of $n-1$ consecutive additions:

$$S = (((\dots(a_1 + a_2) + a_3) + \dots + a_{n-1}) + a_n.$$

Define

$$\begin{aligned}
S_2 &= fl(a_1 + a_2),\\
S_3 &= fl(S_2 + a_3),\\
S_4 &= fl(S_3 + a_4),\\
&\vdots\\
S_n &= fl(S_{n-1} + a_n).
\end{aligned}$$

Recall the formula

$$fl(x) = x(1+\varepsilon).$$

Summation

$$
\begin{aligned}
S_2 &= (a_1 + a_2)(1 + \varepsilon_2), \\
S_3 &= (S_2 + a_3)(1 + \varepsilon_3), \\
S_4 &= (S_3 + a_4)(1 + \varepsilon_4), \\
&\vdots \\
S_n &= (S_{n-1} + a_n)(1 + \varepsilon_n).
\end{aligned}
$$

$$
\begin{aligned}
S_2 &= (a_1 + a_2)(1 + \varepsilon_2), \\
S_3 &= (S_2 + a_3)(1 + \varepsilon_3), \\
S_4 &= (S_3 + a_4)(1 + \varepsilon_4), \\
&\vdots \\
S_n &= (S_{n-1} + a_n)(1 + \varepsilon_n).
\end{aligned}
$$

Then,

$$
\begin{aligned}
S_3 &= (S_2 + a_3)(1 + \varepsilon_3), \\
&= ((a_1 + a_2)(1 + \varepsilon_2) + a_3)(1 + \varepsilon_3), \\
&\approx (a_1 + a_2 + a_3) + a_1(\varepsilon_2 + \varepsilon_3), \\
&+ a_2(\varepsilon_2 + \varepsilon_3) + a_3\varepsilon_3, \\
S_4 &\approx (a_1 + a_2 + a_3 + a_4) + a_1(\varepsilon_2 + \varepsilon_3 + \varepsilon_4) \\
&+ a_2(\varepsilon_2 + \varepsilon_3 + \varepsilon_4) + a_3(\varepsilon_3 + \varepsilon_4) + a_4\varepsilon_4.
\end{aligned}
$$

And finally we get

$$\begin{aligned} S_n \approx (a_1 + a_2 + \ldots + a_n) &+ a_1(\varepsilon_2 + \ldots + \varepsilon_n) \\ + a_2(\varepsilon_2 + \ldots + \varepsilon_n) &+ a_3(\varepsilon_3 + \ldots + \varepsilon_n) \\ + a_4(\varepsilon_4 + \ldots + \varepsilon_n) &+ \ldots + a_n\varepsilon_n. \end{aligned}$$

And finally we get

$$\begin{aligned} S_n \approx\, &(a_1 + a_2 + \ldots + a_n) + a_1(\varepsilon_2 + \ldots + \varepsilon_n) \\ &+ a_2(\varepsilon_2 + \ldots + \varepsilon_n) + a_3(\varepsilon_3 + \ldots + \varepsilon_n) \\ &+ a_4(\varepsilon_4 + \ldots + \varepsilon_n) + \ldots + a_n\varepsilon_n. \end{aligned}$$

We are interested in the error $S - S_n$:

$$\begin{aligned} S - S_n \approx\, &-a_1(\varepsilon_2 + \ldots + \varepsilon_n) - a_2(\varepsilon_2 + \ldots + \varepsilon_n) - a_3(\varepsilon_3 + \ldots + \varepsilon_n) \\ &- a_4(\varepsilon_4 + \ldots + \varepsilon_n) - \ldots - a_n\varepsilon_n. \end{aligned}$$

And finally we get

$$\begin{aligned} S_n \approx (a_1 + a_2 + \ldots + a_n) + a_1(\varepsilon_2 + \ldots + \varepsilon_n) \\ + a_2(\varepsilon_2 + \ldots + \varepsilon_n) + a_3(\varepsilon_3 + \ldots + \varepsilon_n) \\ + a_4(\varepsilon_4 + \ldots + \varepsilon_n) + \ldots + a_n\varepsilon_n. \end{aligned}$$

We are interested in the error $S - S_n$:

$$\begin{aligned} S - S_n \approx -a_1(\varepsilon_2 + \ldots + \varepsilon_n) - a_2(\varepsilon_2 + \ldots + \varepsilon_n) - a_3(\varepsilon_3 + \ldots + \varepsilon_n) \\ - a_4(\varepsilon_4 + \ldots + \varepsilon_n) - \ldots - a_n\varepsilon_n. \end{aligned}$$

Using the last relation, in order to minimize the error $S - S_n$, we can establish the strategy for summation:
initially rearrange the terms in increasing order:

$$|a_1| \leq |a_2| \leq |a_3| \leq \ldots \leq |a_n|.$$

And finally we get

$$\begin{aligned} S_n \approx (a_1 + a_2 + \ldots + a_n) + a_1(\varepsilon_2 + \ldots + \varepsilon_n) \\ + a_2(\varepsilon_2 + \ldots + \varepsilon_n) + a_3(\varepsilon_3 + \ldots + \varepsilon_n) \\ + a_4(\varepsilon_4 + \ldots + \varepsilon_n) + \ldots + a_n\varepsilon_n. \end{aligned}$$

We are interested in the error $S - S_n$:

$$\begin{aligned} S - S_n \approx -a_1(\varepsilon_2 + \ldots + \varepsilon_n) - a_2(\varepsilon_2 + \ldots + \varepsilon_n) - a_3(\varepsilon_3 + \ldots + \varepsilon_n) \\ - a_4(\varepsilon_4 + \ldots + \varepsilon_n) - \ldots - a_n\varepsilon_n. \end{aligned}$$

Using the last relation, in order to minimize the error $S - S_n$, we can establish the strategy for summation:
initially rearrange the terms in increasing order:

$$|a_1| \leq |a_2| \leq |a_3| \leq \ldots \leq |a_n| .$$

In this rearrangement, smaller numbers a_1 and a_2 will be multiplied with larger numbers $\varepsilon_2 + \ldots + \varepsilon_n$, and a larger number a_n will be multiplied with a smaller number ε_n.

Summation. Example with chopping

Four digit calculator with chopping is being used to compute the following sum:

$$S = \sum_{i=1}^{n} \frac{1}{i} = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}.$$

Number of terms, n	Exact value	SL	Error	LS	Error
10	2.929	2.928	0.001	2.927	0.002
25	3.816	3.813	0.003	3.806	0.010
50	4.499	4.491	0.008	4.470	0.020
100	5.187	5.170	0.017	5.142	0.045
200	5.878	5.841	0.037	5.786	0.092
500	6.793	6.692	0.101	6.569	0.224
1000	7.486	7.284	0.202	7.069	0.417

SL: *smallest to largest* strategy; LS: *largest to smallest* strategy.

Summation. Example with rounding

Four digit calculator with rounding is being used to compute the following sum:

$$S = \sum_{i=1}^{n} \frac{1}{i} = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}.$$

Number of terms, n	Exact value	SL	Error	LS	Error
10	2.929	2.929	0	2.929	0
25	3.816	3.816	0	3.817	-0.001
50	4.499	4.500	-0.001	4.498	0.001
100	5.187	5.187	0	5.187	0
200	5.878	5.878	0	5.876	0.002
500	6.793	6.794	-0.001	6.783	0.010
1000	7.486	7.486	0	7.449	0.037

SL: *smallest-to-largest* strategy; LS: *largest-to-smallest* strategy.

Conclusions:

1. *Smallest-to-largest* strategy is more preferable than the *largest-to-smallest*, since the error in the first strategy is at least twice as small.

Summation

Conclusions:

1. *Smallest-to-largest* strategy is more preferable than the *largest-to-smallest*, since the error in the first strategy is at least twice as small.
2. Rounding is much better that chopping. Just compare the errors in the first and second tables.

Summation

Conclusions:

1. *Smallest-to-largest* strategy is more preferable than the *largest-to-smallest*, since the error in the first strategy is at least twice as small.
2. Rounding is much better that chopping. Just compare the errors in the first and second tables.
3. Observe that in the rounding case, some of the errors are 0, since rounding errors can be either positive or negative, and thus they might cancel each other.