How do we truncate a real number to fit our storage mechanism?

Rounding or Chopping

Suppose $x = .a_1 a_2 \cdots a_n a_{n+1} \cdots a_m$ using m digits.

We round to n decimal places by looking at a_{n+1}.

If $a_{n+1} = 0, 1, 2, 3$ or 4

then $x = .a_1 a_2 \cdots a_n$ (after rounding)

If $a_{n+1} = 5, 6, 7, 8, 9$

then $x = .a_1 a_2 \cdots (a_n + 1)$ (after rounding)

last digit increased by 1

OR: we could chop and simply discard $a_{n+1} \; a_{n+2} \cdots a_m$

So $x = .a_1 a_2 \cdots a_n$

To quantify error we have

Absolute error $|p - p^*|$

Relative error $\frac{|p - p^*|}{|p|}$ $p \neq 0$

to measure the error in an approximation p^* to p.

Significant Digits

p^* approximates p to t significant digits if the relative error is less than

$$5 \times 10^{-t}$$

ie t is the largest integer so that

$$\frac{|p - p^*|}{|p|} < 5 \times 10^{-t}$$

Floating Point Arithmetic

Let $fl(x)$ denote the machine representation of x.

If we want to compute $x + y$ then the computer returns

$$fl\left(fl(x) + fl(y)\right)$$

Even these "small" errors lead to problems.

Cancellation Error

(subtracting nearly equal numbers)

Consider

$fl(x) = 0.d_1 d_2 \cdots d_p \alpha_{p+1} \alpha_{p+2} \cdots \alpha_K \times 10^n$

$fl(y) = 0.d_1 d_2 \cdots d_p \beta_{p+1} \beta_{p+2} \cdots \beta_K \times 10^n$

and $x > y$

We have $fl(fl(x) - fl(y))$

$= 0.\sigma_{p+1} \sigma_{p+2} \cdots \sigma_K \times 10^{n-p}$

where

$0.\sigma_{p+1} \sigma_{p+2} \cdots \sigma_K$

$= 0.\alpha_{p+1} \alpha_{p+2} \cdots \alpha_K - 0.\beta_{p+1} \beta_{p+2} \cdots \beta_K$

$\Rightarrow$ Only $K-p$ digits of significance.

We have lost p digits.

Example

$p = 0.54617$

$q = 0.54601$

Exact value $r = p - q = 0.00016$

But now with 4 digit rounding

$p^* = 0.5462$, $q^* = 0.5460$

$r^* = p^* - q^* = 0.002$

and $\frac{|r - r^*|}{|r|} = 0.25$

$\Rightarrow$ 1 significant digit

Even though p^* and q^* are accurate to 4 and 5 significant figures respectively.

Example: Consider $f(x) = \frac{1-\cos x}{x^2}$

Let $\bar{x} = 1.2 \times 10^{-5}$. Then

$c = \cos(\bar{x}) = 0.9999999999$

(rounded to 10 digits)

and $\frac{1-c}{\bar{x}^2} = \frac{10^{-10}}{1.44 \times 10^{-10}} = 0.69\ldots$

which is clearly wrong (see a plot)

To fix, we rewrite as

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

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

Now $f(\bar{x}) = 0.5$ (computationally)

IDEA: Use expression manipulation to remove the cancellation (and the cancellation error).

Another way to reduce the round off error is to reduce the number of floating point operations.

Ex. Polynomial Evaluation using nested multiplication.

$$f(z) = 1.01z^4 - 4.62z^3 - 3.11z^2 + 12.2z - 1.99$$

$$= (1.01z^3 - 4.62z^2 - 3.11z + 12.2)z - 1.99$$

$$= [(1.01z^2 - 4.62z - 3.11)z + 12.2]z - 1.99$$

$$= \{[(1.01z - 4.62)z - 3.11]z + 12.2\}z - 1.99$$

Example.

Solve for x: $ax^2 + bx + c = 0$

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Say $b = 600$, $a = c = 1$

- What could go wrong?
- How could we reformulate the problem?
- What should we do if b was -600?