Example 5

Algebraically $\frac{a \cdot b}{c} = \frac{(a \cdot b)}{c} = \left(\frac{a}{c}\right) \cdot b = a \cdot \left(\frac{b}{c}\right)$

What happens if we compute $\left(\frac{a}{c}\right) \cdot b$ using floating point?

$\left(\frac{a}{c}\right) = \frac{1.34}{7.99} = 0.167709637 \xrightarrow{\text{chop}} 0.167$

$\left(\frac{a}{c}\right) \cdot b = (0.167) \cdot (0.362) = 0.060454 \xrightarrow{\text{chop}} \boxed{0.0604}$

Note that

$$\frac{(a \cdot b)}{c} \neq \left(\frac{a}{c}\right) \cdot b$$

though both answers are close to each other

Example 6

compute the solution to

$$\begin{aligned} 113x + 114y &= 1 \\ 114x + 113y &= -1 \end{aligned}$$

exact soln is $x = -1$, $y = 1$

using 2 digits with chopping

NOTE we can only use 2 digits, so right away, we make a storage error when the set of equations is stored on a computer

$113 \longrightarrow 110$

$114 \longrightarrow 110$

on a computer this looks like

$$\left.\begin{aligned} 110x + 110y &= 1 \\ 110x + 110y &= -1 \end{aligned}\right\} \text{ no solution}$$

important → one consequence of floating point computing is that a problem that has a solution can be changed into a problem that has no solution

Example 7

What if we use 3 digits with chopping in the previous example?

In this case, we don't have the storage error, so the computer sees

$$113x + 114y = 1 \quad ①$$
$$114x + 113y = -1 \quad ②$$

There are many ways to solve this. For this example, solve ① for x and substitute this into ②.

$$113x + 114y = 1$$
$$113x = 1 - 114y$$
$$x = \frac{1}{113} - \frac{114}{113}y$$

$\frac{1}{113} = 0.00884955\ldots \xrightarrow{\text{chop}} = 0.00884$

$\frac{114}{113} = 1.00884955\ldots \xrightarrow{\text{chop}} = 1.00$

The floating point version of x is

$$x = 0.00884 - 1.00y$$

Substitute into equation ②

$$114(0.00884 - 1.00y) + 113y = -1$$
$$114(0.00884) - 114(1.00y) + 113y = -1$$
$$114(0.00884) = 1.00776 \xrightarrow{\text{chop}} 1.00$$
$$114(1.00y) = 114y$$
$$1.00 - 114y + 113y = -1$$
$$-y = -2.00 \rightarrow \boxed{y = 2.00}$$

$$x = 0.00884 - 1.00(2.00) = -1.99116 \xrightarrow{\text{chop}} -1.99$$

$$\boxed{x = -1.99}$$

compute the relative errors.

$$Rel_x = \frac{-1.99 - (-1)}{-1} = 0.99$$

$$Rel_y = \frac{2.00 - 1}{1} = 1.00$$

In this example, a solution could be computed, but the errors are so large that the solution is useless.

Do things improve if we use 4 digits? t/w problem

Making Things more concrete

computers use scientific notation to represent floating point numbers

Example 8

$$13127.69248 \rightarrow \underbrace{1.3127692 48}_{\text{mantissa}} \times \underbrace{10}_{\text{base}}^{4} \leftarrow \text{exponent}$$

A particular computer architecture is defined by the following parameters

1) β = base for computation (usually 2, sometimes 10 or 16)

2) t = precision; this is the number of digits in the mantissa

3) exponent range; e = smallest exponent allowed
E = largest exponent allowed

In addition, the first digit of the mantissa must be non-zero (this is called the normalizing rule)

Example 9

Hand calculator $\quad \beta = 10$
$t = 10$ or 12
$e = -99$
$E = 99$

Specifying values of β, t, e and E sets up a template that all floating point values must fit into.

Example 10 Suppose we have a computer with

$\beta = 10$
$t = 3$
$e = -4$
$E = 4$

- What is the largest number that can be represented on this computer?

$\square$ ← must be between -4 and 4

$\_ . \_\_ \times 10^{\square}$

3 digits in mantissa

answer: $9.99 \times 10^{4} \longrightarrow 99900$

- What is the smallest number we can represent?

$\_ . \_\_ \times 10^{\square}$

answer: $1.00 \times 10^{-4} \longrightarrow 0.000100$

These 2 examples illustrate that there is a limit to the range of numbers that can be represented

- can we represent every number between 0.000100 and 99900 on this computer?

answer: No For example, we cant represent
84329 (must limit to 3 digits)

$84329 \rightarrow 8.43 \times 10^4$

This means that there are gaps in the numbering system. A consequence of this is that there are many real numbers (in the mathematical sense) that have the same floating point representation

- What happens if we try to store 173924?

$173492 \longrightarrow 1.73492 \times 10^5$
$\longrightarrow 1.73 \times 10^5$

answer: The exponent is too large. This is called overflow. This is a very bad thing to have happen.

- What happens if we try to store 0.000027649?

$0.0000\,27649 \longrightarrow 2.7649 \times 10^{-5}$
$\longrightarrow 2.76 \times 10^{-5}$

answer: The exponent is too small. This is called underflow. This is also bad, but not as bad as overflow.