SUMMARY  Floating Point Arithmetic
CURRENT READING  Burden & Faires Sections 1.2 and 1.4

Chopping versus Rounding
Let’s consider the relative error involved in using chopping versus rounding in the representation of machine numbers with \( k \) decimal digits.

Any positive real number \( y \) can be normalized to be written in the form

\[
y = 0.d_1d_2d_3 \cdots d_kd_{k+1} \cdots \times 10^n
\]

This is called a floating point number. It is clear that if one restricts the number of digits which can be used to represent the floating point number one will have to either perform chopping or rounding in order for the floating point number to “fit” the allotted memory of \( k \) digits.

Exercise
Write down an expression which represents \( \epsilon_{rel} \), the relative error involved in representing \( y \) as a decimal machine number \( fl(y) \) (i.e. a floating point number using a fixed number of decimal digits).

Example
(a) Write down what \( fl(y) \) would look like if chopping were implemented

(b) Write down what \( fl(y) \) would look like if rounding were implemented
**Groupwork**

Show that the expression involving $k$ which gives you an upper bound for the relative error involved in using chopping arithmetic is $\epsilon_{rel} = 10^{-k+1}$.

**Example**

It can be shown that a bound for the relative error involved in using rounding arithmetic is half that for chopping, $\epsilon_{rel} = 0.5 \times 10^{-k+1} = 5 \times 10^{-k}$.

**Groupwork**

Compute the following arithmetic operations using finite precision arithmetic.
Half the group should use five-digit chopping arithmetic and the other half five-digit rounding arithmetic.
Also compute the absolute and relative error from the “actual value.”
What results do you expect?

(a) $\frac{1}{5} \oplus \frac{5}{7} =$

(b) $\frac{1}{5} \ominus \frac{5}{7} =$

(c) $\frac{1}{5} \ominus \frac{5}{7} =$

(d) $\frac{1}{5} \ominus \frac{5}{7} =$

**ANNOUNCEMENTS**

Remember: Class is cancelled on Monday September 7 and Friday September 11 next week.