Models for arithmetic by computer.

Last time, we defined interval arithmetic and showed that calculations with slightly fuzzy numbers are... weird.

We're now going to introduce two more models which get closer to what a computer does.

Definition. (Base-10, n-digit ~~floating~~ fixed point)
We define fixed point numbers to have at most n digits to the right of the decimal point.

round up!

$fx(x, n)$ = the closest* number to x
in the fixed point #s with
n-digits.

Note. $fx(x,n)$ is always rational, in fact always an integer multiple of 10^{-n}.

We define the fixed point arithmetic operations by

$$x \odot y = fx(fx(x,n) \odot fx(y,n))$$

where $\odot$ is one of $+, -, \times, \div$.

~~Examples~~. We observe that the results of the fixed point operations are not always correct!

Example. Let $n=5$.

$$100 \times 10^{-6} = 10^{-4}$$

$$fx(fx(100,5) \times fx(10^{-6},5)) = fx(100 \times 0) = fx(0) = 0.$$

Example. Let $n=4$.

~~$0.1036 \times 0.2122 =$~~

$$0.2122 \times 0.2081 = 0.044158 \xrightarrow{fx} 0.0442$$

$$0.1036 \times 0.4247 = 0.04399 \xrightarrow{fx} 0.0440$$

We can carry this one step further

$$(0.1036 \times 0.4247 - 0.2122 \times 0.2081)$$

$$\Big\downarrow \text{multiply and round}$$

$$(0.0440 - 0.0442)$$

$$\Big\downarrow \text{subtract and round}$$

$$-0.0002$$

The correct answer is -0.0001599. The difference might not look like much, but it's almost 30% of the correct answer!

The difference between x and $fx(x,n)$ is called roundoff error.

Definition. If a is correct and b is an approximation to a, then

<u>absolute</u> error in b is $|b-a|$

<u>relative</u> error in b is $\dfrac{|b-a|}{|a|}$

Observation. The <u>absolute</u> roundoff error $|x - fx(x,n)| \leq \frac{1}{2} \times 10^{-n} = 5 \times 10^{-n-1}$.
The <u>relative</u> roundoff error is unbounded! (Just make x very small.)

(precision_linear_equations_slideshow.nb)

Fixed point arithmetic seems to be something you could program, but there's still a problem - we haven't bounded the number of digits left of the decimal point. This leads to a natural idea - bound the total number of digits by writing everything in scientific notation.

Definition. (Base 10 n-digit ~~fixe~~ floating point) A number in n-digit floating point is written $\pm d_1 . d_2 \cdots d_n \times 10^e$ where e is an integer between bounds $m \leq e \leq M$, and $d_1 \neq 0$ or all $d_i = 0$.

Note that all these are rational numbers and that the set is weird.

Example. 1-digit with $0 \leq e \leq 1$.

The numbers are $\pm$

1 2 3 4 5 6 7 8 9

10 20 30 40 50 60 70 80 90

We define $fl(x)$ to be the nearest floating point number to x in the current system, *(round up for ties) and

$$x \odot y = fl(fl(x) \odot fl(y))$$

where $\odot$ is one of $+, -, *, \div$.

Example.

$$2+5 = fl(fl(2)+fl(5)) = fl(7) = 7.$$

$$2+30 = fl(fl(2)+f(30)) = fl(32) = 30. \quad (?!?)$$

$$3 \times 6 = fl(fl(3) \times fl(6)) = fl(18) = 20. \quad (?!!)$$

$$24 \div 5 = fl(fl(24) \div fl(5)) = fl(20 \div 5) = fl(4) = 4.$$

(This last one is particularly disturbing because it's not even $fl(\frac{24}{5}) = fl(4.8) = 5$.)

Even in this number system, we can see several problems are going to happen:

$$30 \times 5 = fl(30 \times 5) = fl(150) = ?$$

We could take the definition of fl literally and round to 100, but computers instead signal an <u>overflow</u> error (and stop computing) when the result is larger than the largest representable number.

$$2 \div 30 = fl(1/16) = ?$$

Again, we could round (to 0) but computers instead signal <u>underflow</u> when the result is between 0 and the smallest nonzero representable number.

Floating point is better than fixed point, but there are still error problems.

~~Proposition. The absolute error~~ $\cancel{|x - fl(x)|}$ ~~is bounded by~~ $\cancel{\frac{1}{2} \times 10^{M-n}}$. ~~The relative error~~

$$\cancel{\frac{|x - fl(x)|}{|x|}} \cancel{\leq \frac{1}{2} \times 10^{m-n}}$$

~~Proof. We know that any~~ x ~~between~~ $\cancel{-9.9\ldots9 \times 10^{M}}$ ~~and~~ $\cancel{9.9\ldots9 \times 10}$

We say that x is <u>in range</u> if

$$-\underbrace{9.9\ldots9}_{n \text{ digits}} \times 10^M \leq x \leq -\underbrace{1.000}_{n \text{ digits}} \times 10^m$$

or

$$\underbrace{1.0\ldots0}_{n \text{ digits}} \times 10^m \leq x \leq \underbrace{9.9\ldots9}_{n \text{ digits}} \times 10^M.$$

(That is, if x does not trigger an underflow or overflow error.)

Proposition. If x is in range, then

$$|x - fl(x)| \leq \frac{1}{2} 10^{M-n+1}$$

Proof. Since x is in range, it can be written as a (possibly infinite decimal)

$$x = \pm d_1 . d_2 \ldots d_n d_{n+1} \ldots \times 10^e$$

where $d_1 \neq 0$ and $m \leq e \leq M$. ~~Thus~~

Rounding $\cancel{x}$ $d_1 . d_2 \ldots d_n d_{n+1} \ldots$ to n digits changes its value by at most $\frac{1}{2} 10^{-n+1}$. This is multiplied by 10^e, so

$$
\begin{aligned}
|x - fl(x)| &\leq \frac{1}{2} 10^{-n+1} \times 10^e \\
&\leq \frac{1}{2} 10^{e-n+1} \\
&\leq \frac{1}{2} 10^{M-n+1}. \qquad \square
\end{aligned}
$$

Exercise. What's $fl(9.9\ldots96 \times 10^e)$? Does it still obey the proposition?

Proposition. If x is in range, then

$$\frac{|x - fl(x)|}{|x|} \leq \frac{1}{2} 10^{-n+1}$$

Proof. Again,

$$x = \pm d_1 . d_2 \ldots d_{n\ldots} \times 10^e$$

and

$$|x - fl(x)| \leq \frac{1}{2} 10^{e-n+1}$$

Now $|x| \geq 1.0\ldots0 \times 10^e$, so

$$\frac{|x - fl(x)|}{|x|} \leq \frac{1}{2} 10^{-n+1} \qquad \square$$

Note: The absolute error depends on M (and may be large!), while the relative error depends only on n (and is quite small).

Corollary. If x is in range,

$$fl(x) = x(1+\delta) \qquad \left(|\delta| \leq \frac{1}{2} 10^{-n+1}\right)$$

Proof. Let

$$\delta = \frac{fl(x) - x}{x}.$$

(Simplifying, we have

$$x\left(1+\frac{fl(x)-x}{x}\right)=fl(x).$$)

We just proved $\left|\frac{fl(x)-x}{x}\right| \leq \frac{1}{2}10^{-n+1}$.