

Floating-point systems

There are infinitely many real numbers, so we can't represent all of them with a fixed number of bits. So let's select a set of real numbers and use them to approximate a range of real numbers.

A floating-point system of representation uses some fixed number of digits p in some base β .

Let's look at the system for the parameters $p = 3$ and $\beta = 10$.

In it, we would likely approximate π by 3.14, and $1/3$ by .333.

To distinguish between say .333 and 333, we need to store the location of the decimal point as well as the digits. The location of the decimal point specifies a scaling of the number by some integer power of 10: $3.14 = 3.14 \times 10^0$, $.333 = 3.33 \times 10^{-1}$ and $333 = 3.33 \times 10^2$.

In general, a number in our system will be of the form:

$$\pm d_0.d_1 \dots d_{p-1} \times \beta^e.$$

The digits portion is called the mantissa, aka significand, and the integer e is called the exponent.

The size of the exponent must be limited in order to represent it in a fixed number of bits: so a floating-point system has two more parameters e_{min} and e_{max} , limiting e to $e_{min} \leq e \leq e_{max}$.

Notice that different representations may correspond to the same number. For example, $2.5 = 2.50 \times 10^0 = 0.25 \times 10^1$. To make the representation unique, we use only what are called normalized representations: ones without leading zeros, i.e. ones with $d_0 \neq 0$.

To continue our example with $p = 3$ and $\beta = 10$, let's suppose the range of exponents is $[-3, 4]$.

The smallest positive number we can represent is then $1.00 \times 10^{-3} = 0.001$, and the largest number we can represent is $9.99 \times 10^4 = 99900$.

Another example: $p = 3$, $\beta = 2$, $e_{min} = -1$ and $e_{max} = 2$. The numbers in this system are

$$\pm (1.d_1d_2)_2 \times 2^e$$

where d_1, d_2 are 0 or 1, and e is $-1, 0, 1$ or 2 . Since the base is 2, d_1 is the $1/2$'s digit, and d_2 is the $1/2^2 = 1/4$'s digit. [In class we drew a number line with all the positive numbers in this system.]

Notice that the larger numbers are spaced out more: the difference between $1.00_2 \times 2^0$ and $1.01_2 \times 2^0$ is 0.25, but between $1.00_2 \times 2^2$ and $1.01_2 \times 2^2$ it's 1. Notice however that in these two examples the percentage change is the same: 25%. Percentage is a relative measure, and the fact that it's so commonly used is an indication that in most numerical work it's relative amounts that matter.

Rounding

Arithmetic with numbers in a floating-point system quickly produces numbers that aren't representable in the system. For example if $\beta = 10$, $1/3$ is not representable. So we choose a floating-point number to approximate the actual number.

The mapping of real numbers to floating-point numbers is called rounding. There are various systems of rounding, some examples are: round towards nearest, round towards zero, banker's rounding. Often people just say "rounding" to mean round towards nearest.

If we represent a number larger in size than the normalized numbers we say there is an overflow. Similarly, representing a non-zero number smaller in size than the normalized numbers is called underflow. Situations involving overflow or underflow often require special handling.

Consider again a floating-point system in which $\beta = 10$ and $p = 3$. Rounding 3.14159 to the nearest floating-point number gives 3.14×10^0 . How far off is the rounded value? The absolute error is one measure: $|3.14 \times 10^0 - 3.14159| = 0.00159$.

Relative error

The difference between 1 and 1.1 is usually more significant than the difference between 100 and 100.1. Relative error is a way of capturing this: it takes the absolute error relative to the actual number:

$$\left| \frac{x_{rep} - x}{x} \right|.$$

The relative error in representing 1.1 by 1 is

$$\left| \frac{0.1}{1} \right| = 0.1 = 10\%$$

and the relative error in representing 100.1 by 100 is

$$\left| \frac{0.1}{100} \right| = 0.001 = 0.1\%.$$

Consider again a system with $\beta = 10$ and $p = 3$. The relative errors in rounding 3.14159 and 31.4159 are both the same:

$$\begin{aligned} \left| \frac{3.14159 \times 10^0 - 3.14 \times 10^0}{3.14 \times 10^0} \right| &= \frac{3.14159 - 3.14}{3.14} \\ \left| \frac{3.14159 \times 10^1 - 3.14 \times 10^1}{3.14 \times 10^1} \right| &= \frac{3.14159 - 3.14}{3.14} \end{aligned}$$

and

$$\frac{0.00159}{3.14} \approx 5 \times 10^{-4} = 0.05\%.$$

Relative error in round to nearest

For non-zero numbers with neither underflow nor overflow, how much relative error can be produced by rounding?

To get a feel for how rounding works, consider a system where β is even and a number

$$d_0.d_1 \dots d_{p-1}d_p \dots \times \beta^e$$

with $d_0 \neq 0$. The number gets rounded down if and only if $d_p \leq \beta/2 - 1$, and the rounded version is

$$d_0.d_1 \dots d_{p-1}00 \dots \times \beta^e$$

with an absolute error of

$$0.0 \dots 0d_p \dots \times \beta^e$$

which is less than

$$0.0 \dots 0 \left(\frac{\beta}{2}\right) 0 \dots \times \beta^e = \beta^{-p} \cdot \frac{\beta}{2} \cdot \beta^e$$

so the relative error is less than

$$\begin{aligned} \frac{\frac{\beta^{-p+1}}{2} \beta^e}{d_0.d_1 \dots d_{p-1}d_p \dots \times \beta^e} &\leq \frac{\frac{\beta^{-p+1}}{2}}{1.0 \dots} \\ &= \frac{\beta^{-p+1}}{2}. \end{aligned}$$

Requiring β to be even makes what's happening with the digits more obvious, but is not necessary. Let's make a more general argument.

A number

$$d_0.d_1 \dots d_{p-1}d_p \dots \times \beta^e$$

with $d_0 \neq 0$ gets rounded to either

$$d_0.d_1 \dots d_{p-1} \times \beta^e$$

or

$$d_0.d_1 \dots (d_{p-1} + 1) \times \beta^e$$

(the second number may have to be represented differently if $d_{p-1} + 1 = \beta$, but its value stays the same). The two numbers we may round differ by

$$d_0.d_1 \dots (d_{p-1} + 1) \times \beta^e - d_0.d_1 \dots d_{p-1} \times \beta^e = \beta^{-(p-1)},$$

so if we round to the nearest then we round by at most half this difference: $\frac{\beta^{-p+1}}{2} \times \beta^e$. The relative error is thus at most

$$\frac{\beta^{-p+1}/2 \times \beta^e}{1.0\dots \times \beta^e} = \frac{\beta^{-p+1}}{2}.$$

For our example with $\beta = 2$ and $p = 3$, rounding produces a relative error of at most $2^{-3+1}/2 = 1/8 = 12.5\%$.

Accumulation of error

Consider again a system with $\beta = 10$ and $p = 3$. The relative error in rounding is at most $10^{-3+1}/2 = 0.5\%$, but what if we perform operations on the rounded numbers?

Consider $b^2 - 4ac$ for $b = 3.34$, $a = 1.22$ and $c = 2.28$. The exact value is 0.0292, which is a number in the system. But when a machine evaluates the subexpressions in our system, it produces

$$\begin{aligned} b^2 &= 3.34^2 \\ &= 11.1556 \\ &\text{which rounds to } 11.2 \end{aligned}$$

and

$$\begin{aligned} 4ac &= 4 \times 1.22 \times 2.28 \\ &= 4.88 \times 2.28 \\ &= 11.1264 \\ &\text{which rounds to } 11.1 \end{aligned}$$

and then $b^2 - 4ac$ is approximated by $11.2 - 11.1 = 0.1$.

This is a relative error of

$$\left| \frac{0.0292 - 0.1}{0.0292} \right| > 2.4 = 240\%.$$

Generally, subtraction of two relatively close numbers (or addition of a number and one close to its negative) can lead to large relative errors. The subtraction produces a number much smaller than the two numbers, and the errors which were small compared to the two numbers can be big compared to the difference. This is called catastrophic cancellation.

Consider calculating the roots of

$$ax^2 + bx + c = 0 \quad (a \neq 0)$$

with the formulae:

$$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

The cancellation in $b^2 - 4ac$ can introduce a large relative error *during* the calculation of the above formulae, but it doesn't necessarily cause a large relative error in the final result. Catastrophic cancellation can only occur if $b^2 - 4ac$ is small relative to b^2 . Then $\sqrt{b^2 - 4ac}$ is small relative to $-b$. Even a somewhat large relative error in $\sqrt{b^2 - 4ac}$ might not be a problem relative to $-b$.

With the values of a, b and c above we would calculate r_1 as

$$\begin{aligned} \frac{-b + \sqrt{b^2 - 4ac}}{2a} &\approx \frac{-b + \sqrt{0.1}}{2a} \\ &\approx \frac{-3.34 + 0.316}{2 \cdot 1.22} \\ &\approx \frac{-3.02}{2.44} \\ &\approx -1.24. \end{aligned}$$

Without the rounding in $b^2 - 4ac$ we would calculate

$$\frac{-3.34 + \sqrt{0.0292}}{2 \cdot 1.22}$$

and though $\sqrt{0.0292} \approx 0.171$ shows that 0.316 has a large relative error, that relative error is small compared to -3.34 :

$$\begin{aligned} \frac{-3.34 + 0.171}{2 \cdot 1.22} &\approx \frac{-3.17}{2.44} \\ &\approx -1.3. \end{aligned}$$

The final relative error here is about 5%.

The real problem with the formulae above is the potential catastrophic calculation when combining $-b$ and $\sqrt{b^2 - 4ac}$. A catastrophic cancellation can occur if $|b|$ is much larger than $4ac$, since then b is close to one of $\pm\sqrt{b^2 - 4ac}$. The division by $2a$ doesn't help: it's a scaling, and relative error is (essentially designed to be) independent of scaling (for example, 101 is 1% more than 100, and 50.5 is 1% more than 50).