

In-class exercises, Sept. 25

Due on Tuesday, Oct. 1 on plain old paper.

Binary floating-point numbers have the form

$$\hat{x} = \pm a_0.a_1a_2\cdots a_d \times 2^e.$$

If $\hat{x} \neq 0$, then \hat{x} is *normalized* if $a_0 = 1$. Normalization makes e unique, since it must be the largest integer such that $2^e \leq |\hat{x}|$. If $\hat{x} = 0$, then we may simply define the normalized representation to be 0×2^0 . In IEEE single-precision arithmetic, $d = 23$ (i.e., 24-bit precision), and in double-precision arithmetic, $d = 52$ (i.e., 53-bit precision).

Floating-point arithmetic is *correctly rounded* if the operation $\hat{x} \odot \hat{y}$ is performed *as though* infinite precision were available, followed by rounding. Here \odot refers to the rounded floating-point operations of addition, subtraction, multiplication, and division. All four elementary operations in IEEE arithmetic are correctly rounded, as is the square root operation. In lecture, we discussed the following theorem:

Let \hat{x} and \hat{y} be floating-point numbers of the same precision and range. Suppose that $\hat{x} \cdot \hat{y}$ is defined and representable, where \cdot refers to one of addition, subtraction, multiplication, or division. Let $\hat{x} \odot \hat{y}$ denote the corresponding correctly-rounded result (to the same precision as \hat{x} and \hat{y}). Then there is a real number δ such that

$$\hat{x} \odot \hat{y} = (\hat{x} \cdot \hat{y})(1 + \delta), \quad (1)$$

where $|\delta| \leq \varepsilon$ (the machine epsilon) in all rounding modes and where $|\delta| \leq \frac{1}{2}\varepsilon$ when rounding to nearest.

By defined and representable, we mean that $\hat{x} \cdot \hat{y}$ is a real number (e.g., we aren't dividing by 0) and can be approximated as a floating-point number without overflow or underflow.

1. Consider IEEE single precision.
 - (a) Show that the values 1.0, 2.0, and 3.0 can be represented exactly.
 - (b) You can extend your argument in (a) to show that 4.0, 5.0, 6.0, \dots , M have exact representations. What is M ?
 - (c) Repeat your calculation in (b) for IEEE double precision.

2. Show that the above theorem can be restated in the following terms: In correctly rounded arithmetic, the relative error in the operation $\hat{x} \odot \hat{y}$ never exceeds ε in any rounding mode and never exceeds $\varepsilon/2$ when rounding to nearest.
3. Although the theorem is the best that one can do with floating-point arithmetic of fixed precision and range, rounding errors can accumulate in a chain of calculations. *Catastrophic cancelation* occurs when two floating-point numbers, each of which is the result of a chain of computations and is of comparable size, are subtracted. In this case, the most significant digits disappear and expose the effects of previous roundings. Consider the quadratic formula:

$$r_{\pm} = \frac{-b \pm D}{2a}, \quad \text{where } D = \sqrt{b^2 - 4ac}. \quad (2)$$

What is the relative error in r_+ in terms of the machine epsilon in 6-digit, correctly-rounded decimal arithmetic, when $a = 0.414235$, $b = 12$, and $c = 0.0129134$?

Use Python, Maple, or a pocket calculator. Round the result of each calculation to the nearest 6-digit floating-point number *before* continuing to the next one. (You're trying to simulate the behavior of a computer whose registers hold 6-digit decimal values.) For instance, you can compute

$$\begin{aligned} 4 \otimes a & \models x_1 \\ x_1 \otimes c & \models x_2 \\ b \otimes b & \models x_3 \\ x_3 \ominus x_2 & \models x_4, \end{aligned}$$

and so on. Here \models refers to the truncation to 6 digits by rounding to nearest.

4. (a) Show that

$$r_{\pm} = \frac{2c}{-b \pm \sqrt{b^2 - 4ac}} \quad (3)$$

is equivalent to the usual quadratic formula for the zeroes of the polynomial $ax^2 + bx + c$.

- (b) Repeat the previous problem using Eq. (3). Which of the two equations produces the most accurate value of r_+ ? r_- ? Explain why. (Hint: where does catastrophic cancelation occur?)