Measuring error and cancellation

Jonathan Goodman

January 26, 1999

1 Measuring error

Much of the discussion of error, especially errors induced by inexact computer arithmetic (roundoff error), hinges on how errors are measured. Therefore, we begin with some general observations about measurement and calibration of errors. None of these observations is a theorem without exception.

Usually we are interested in relative rather than absolute error. Suppose A is the exact answer and B is the answer reported by the computer. We usually more interested in the relative error,

$$\textit{relative error} = \frac{B - A}{|A|} \quad , \tag{1}$$

than in the absolute error,

$$\textit{absolute error} = B - A \quad .$$

For example, suppose you are calculating the radius of a hydrogen atom. Is an absolute error of $1\text{Å} = 10^{-8}$cm satisfactory? It seems very small. In light of the fact that the actual radius is about .5Å, this is a 100% error, which is probably unacceptable. On the other hand, an error of \$10M would be acceptable if you are estimating the United States national debt.

A caution is that if A is zero or nearly so this definition may give trouble. For example, suppose we write a program to approximate $A(x) = \cos(x)$. For input x, the computer produces $B(x)$. We probably want accurate answers over a range of x values, say $|x| \leq L = 10$. Then we could ask for

$$\textit{max rel error} = \max_{|x| \leq L} \frac{|B(x) - A(x)|}{|A(x)|} \quad .$$

However, for $x = \pi/2$, $A(x) = 0$. Unless $B(x)$ is exactly zero also (unlikely but not impossible), we would get an infinite maximum relative error. A more sensible measure might be

$$\frac{\max_{|x| \leq L} |B(x) - A(x)|}{\max_{|x| \leq L} |A(x)|} \quad . \tag{2}$$

There are other sensible possibilities as well.

A general principle for more complicated error measures, such as (2), is that they should be dimensionless. The simple (pointwise) relative error measure (1) has this property. The seamingly sensible measure

$$\frac{\max_{|x|\leq L} |B(x) - A(x)|}{\int_{|x|\leq L} |A(x)|\, dx}$$

is not dimensionless, while

$$\frac{\max_{|x|\leq L} |B(x) - A(x)|}{\frac{1}{L}\int_{|x|\leq L} |A(x)|\, dx}$$

is.

2 A model of floating point arithmetic

The basic (IEEE) rule of floating point arithmetic is: "the exact answer correctly rounded". If the number, x, is within the range of floating point numbers[1] then the nearest floating point number to x is

$$\tilde{x} = x(1+\epsilon) \quad \text{where } |\epsilon| \leq \epsilon_{mach}. \tag{3}$$

This ϵ_{mach} is pronounced "machine epsilon". The terminology is left over from the bad old days when every computer had different floating point conventions and therefore a different ϵ_{mach}. A reformulation of (3) is that the relative error induced by a single roundoff operation is not larger than machine epsilon. In single precision, ϵ_{mach} is $2^{-23} \approx 10^{-8}$, corresponding to the fact that there are 23 fraction bits. People often use as a rule of thumb that single precision arithmetic produces roughly seven decimal digits of accuracy (representing the accumulation of several roundoff errors), as long as we don't do too much of it. For double precision the relative error is on the order of 10^{-15}, roughly twice the accuracy, if you measure accuracy by the number of correct digits (i.e. relative error).

Thus, a single floating point rounding produces high relative accuracy no matter how large the number being rounded is, as long as it is within the range of normal floating point numbers. Both the hydrogen atom radius and the US national debt are within the range of single precision numbers.

The working scientific computer (the person) usually keeps (3) as a model of the effect of finite precision arithmetic and otherwise forgets the details of IEEE floating point arithmetic. This could be misleading in rare cases of accidental cancellations of error. A more subtle refinement of (3) comes into play when we do a very long computation. In principle, from (3), with $\epsilon_{mach} \approx 10^{-8}$, we could get $O(1)$ relative errors after just 10^8 sequential computations, even without cancellation (see below). It is almost (but not completely) impossible to make this happen on a computer. We get a more accurate picture of the effect of many rounding errors by assuming that the errors are independent random variables of relative size ϵ_{mach}

[1] That means that $|x|$ is smaller than the largest floating point number and $|x|$ is larger than the smallest normalizable positive floating point number. Denormalized numbers do not behave this way.

and mean value zero[2]. In this picture, the sum of N rounding errors of size ϵ_{mach} will produce a total error closer to $\sqrt{N}\epsilon_{mach}$ than to $N\epsilon_{mach}$. For single precision, this would allow us to do 10^{15} computations rather than 10^8 before "loosing it completely", a much more satisfying number at present computer speeds. Of course, this picture is rough at best. It becomes completely inappropriate in the presence of cancellation or unstable algorithms.

3 Cancellation

"Cancellation" is the main way roundoff error leads to large relative errors. This can happen all at once, in "catastrophic cancellation", or, more commonly, more slowly as errors are amplified through many steps of an unstable computational algorithm. Catastrophic cancellation occurs when we subtract two nearly equal numbers. Suppose x and y are approximated at some point in a computation by $\tilde{x}$ and $\tilde{y}$ with relative accuracy

$$\frac{|x-\tilde{x}|}{|x|} = \epsilon_x \quad \text{, and } \frac{|y-\tilde{y}|}{|y|} = \epsilon_y \quad .$$

Now suppose that $z = x - y$ and we compute $\tilde{z} = \tilde{x} - \tilde{y}$ without any additional errors. Then

$$\frac{|\tilde{z}-z|}{|z|} \leq \frac{|\epsilon_x + \epsilon_y|}{\frac{|x-y|}{|x|+|y|}} \quad ,$$

and it is easy to arrange for equality by choosing the signs of $x - y$, ϵ_x, and ϵ_y. We would be comfortable with $\epsilon_z \sim \epsilon_x + \epsilon_y$ (the best we could hope for). However, the actual ϵ_z could be larger than $\epsilon_x + \epsilon_y$ by a factor of

$$\frac{|x|+|y|}{|x-y|} \quad ,$$

which could be quite large.

Just to repeat, most loss of accuracy is not due to a single catastrophic cancellation, but due to a long series of cancellations that individually do not seem large. An algorithm that allows such a gradual buildup of error is called unstable. We usually diagnose such algorithms by pretending that they work in exact arithmetic and checking how sensitive the stages of the computation are to small perturbations.

[2]We are just as likely to round up as to round down.