COMPUTER SCIENCE

UNIVERSITY OF CALIFORNIA
BERKELEY

THE ERROR-ANALYST'S QUANDARY

W. Kahan

Technical Report 8

August 1972

TECHNICAL REPORT

The Error-Analyst's Quandary [†]

W. Kahan
University of California at Berkeley

August 1972

Numerical analysts often pass the buck by alleging that certain computational schemes are "numerically stable" even when they produce palpably wrong answers. This note is intended first to help the layman understand why those allegations, however misleading, may be true, and second to show numerical analysts that the buck is not so easy to pass as might at first appear.

Suppose you want to compute

$$y = f(x)$$

but your computer gives you z instead and says that

$$z + \Delta z = f(x + \Delta x)$$

for some suitably small Δz and Δx . Can you conclude that z is close to y ? Not necessarily. None the less, such a calculation may be regarded as "stable"; the discrepancy between y and z , if large, will then be blamed upon an "ill-conditioned" function f .

Here is an example. Say $x = \binom{\alpha}{\beta}$ and $f(x) = \alpha - \beta$. Try $\alpha = 1.000$ and $\beta = .9999$ on a 4-significant decimal computer built like some that I have learned to live with:

C1.	C2.	C3.
1.000	1.000	1.000
-0.999 9000	-0.999 ~~0~~	-0.999 9000
0.000 1000	0.001	0.000 ~~1000~~
↙	↙	↙
1.000×10^{-4}	1.000×10^{-3}	0.

Three different answers from three different machines. In each case, however, the computed value z is very nearly, nay, exactly what would have resulted from the exact calculation of f at a slightly perturbed argument $x + \Delta x$:

[†] This research was supported by a grant from the U.S. Office of Naval Research, contract number N00014 69 A 0200 1017

C1	C2	C3
1.000	1.0009	0.9999
-0.9999	-0.9999	-0.9999
0.0001	0.001	0.

Here is another example. Say $f(x) = \sin x$ and $x =$ 31415 92653 58979 32384 62643.38328. Do you really expect to see $y = 0.4971\ldots \times 10^{-6}$? If so, how many significant figures will your computer have to preserve when converting x from decimal to binary, or when dividing x by π ? Perhaps now you see why it is cheaper to produce instead of y a value z which satisfies $z + \Delta z = \sin(x + \Delta x)$ for some Δz amounting to at most a unit or so in z's last place *and* some Δx amounting to perhaps a fraction of a unit in the last retained place of x .

In general, we make a virtue of necessity by saying that a scheme to compute $y = f(x)$ is numerically stable whenever we know small bounds for the perturbations Δz and Δx in the equation $z + \Delta z = f(x + \Delta x)$ satisfied by the computed value z . And if z is then very different from y we pass the blame to f by describing it as "ill-conditioned" at x . In effect, we simplify the problem of estimating $y - z$ by abstracting from a complicated computational scheme just two numbers, the bounds upon Δz and Δx , whence the estimation of $y - x$ reduces to an ostensibly machine-independent analysis of the properties of f .

Unfortunately, the simplification is sometimes complicated by nasty problems. First is the vagueness of our concept of numerical stability. The function f may be regarded as mapping one metric space into another, but the spaces are not always obvious. For example, when $f = \alpha^{\beta}$ should we regard its domain as a two-space of pairs $\binom{\alpha}{\beta}$ or, if we are concerned only with $\beta = 2$, as a one-space of numbers (α) ? More generally, how do we distinguish between those aspects of a problem which are, by association with f , denied any variation, and those aspects which are, by association with x , exposed to slight perturbations? And how should the metrics be chosen? The metrics should ideally reflect the interests of the man who wants to compute $y = f(x)$ by assigning to equally important (or equally insignificant) variations the same measure of magnitude. In practice, the metric tends to reflect mainly the limitations of the equipment or the inclinations of the numerical analyst. Finally, even when the metric spaces are perfectly obvious, we encounter an unavoidable arbitrariness in the bounds upon Δx and Δz, for we can always diminish one at the expense of increasing the other without altering the computational scheme in any way. For example, when $f(x) = \sqrt{x}$ we can validly write $z + \Delta z = \sqrt{x + \Delta x}$ with $\frac{|\Delta z|}{z} \leq \zeta$ and $\frac{|\Delta x|}{x} \leq \xi$ using any bounds ζ and ξ that satisfy both $(1 + \xi) / (1 - \zeta)^2 \geq 1 + \varepsilon$ and $(1 - \xi) / (1 + \zeta)^2 \leq 1 - \varepsilon$ for some $\varepsilon > 0$ that depends upon the scheme's accuracy. More generally, we set $\xi = 0$ for the sake of simplicity whenever we can do so without forcing ζ to be embarrassingly large.

A second nasty problem arises when we try to prove that some scheme is stable. Some familiar schemes, long believed to be stable, have not yet been proved stable. For example, suppose $f(x) = x^{-1}$ for $n \times n$ matrices x with fixed but large n. Nobody has yet obtained bounds for $\|\Delta x\|/\|x\|$ and $\|\Delta z\|/\|z\|$ in

$$z + \Delta z = (x + \Delta x)^{-1}$$

which are simultaneously both independent of x and not exponentially growing functions of n, despite that Gaussian Elimination with pivoting and other comparable techniques are regarded (probably rightly) as stable ways to invert matrices no matter how nearly singular those matrices may be.

A third nasty problem arises when we realize that no computational scheme exists for its own sake; it is a means to an end. And that end is generally reached via a concatenation of schemes. For example, to compute $h(x) = g(f(x))$ we may naturally apply f to x to get y, and then g to y to get $h(x) = g(y)$. But we will not actually get $h(x)$; instead we shall obtain, in place of y, a value z satisfying $z + \Delta' z = f(x + \Delta' x)$ for some small-bounded $\Delta' x$ and $\Delta' z$, and then we shall construct in place of $h(x)$ some u satisfying $u + \Delta'' u = g(z + \Delta'' z) = g(f(x + \Delta' x) - \Delta' z + \Delta'' z)$.
There is no guarantee in general that small perturbations Δu and Δx exist satisfying $u + \Delta u = h(x + \Delta x)$. Thus, the concatenation of two stable schemes could be (and usually is) unstable.

There are theorems which describe some of the circumstances when concatenated schemes are stable. Few of those theorems are both interesting and general. Their gist tends to be of the following kind (for the example $h = f(g)$ above);

In order to compute $h(x) = f(g(x))$ in a stable way, we must ensure that the errors $\Delta' z$ and $\Delta'' z$ in the intermediate result $z \doteq f(x)$ are appropriately correlated, despite that those errors may be astonishingly large without vitiating stability. The appropriate correlations must all too often be described in a way which exhumes just those computational details that the error-analyst had hoped to bury in the course of distilling all computational errors into two simple bounds.

Thus do we perceive the error analyst's quandary; when should the error in a computational scheme be summarized in a simple way? Do so too soon, and the result may be too weak to be useful. Do so too late, and the result may be too complicated to be comprehended. And there is no guarantee that a gap exists between "too soon" and "too late".