Where Numerics Matter…

How Computer Arithmetic Can Lead You Astray

By David Sayers, PhD and John Morrissey, PhD, Numerical Algorithms Group

Even the most powerful computers can and will lead those who do not understand their limitations astray. If it is important to get the right answer, which is of course the assumed minimum of any serious financial engineering application, you must understand and master the nuances of numerics in a computational environment. This is especially true for portfolio managers who employ higher level math packages and who therefore need to understand alternative approaches to errors that arise. In this article, we will examine some of the basic problems inherent in ALL computer arithmetic.

Simple Addition

Consider first the inexactness of a computer for the most basic handling of numbers. Let's say that you are required to write a small program to sum a sequence of positive numbers. The numbers may be presented in ascending order of magnitude, descending order, or in arbitrary order.

$$\sum_{i=1}^{N} a_i, \quad a_i \geq 0, \quad N >> 1$$

Unfortunately, finite machine arithmetic is not necessarily associative:

$$(1 + \delta) + \delta$$

is *not* necessarily equal to

$$1 + (\delta + \delta)$$

if δ is sufficiently small. In fact either expression might give a result of 1, though the second expression will succeed in some cases where the first fails.

Typically, $1 + \delta$ will result in 1 for $\delta < 1.11\,E - 16$.

It turns out that the order in which you add things together can matter significantly.

Another everyday source of computer arithmetic error occurs when the mean of a set of numbers is outside their range. The following example in 3-figure arithmetic is mirrored in the 8 or 16 figure arithmetic of computers:

$$(5.01 + 5.03)/2 = 10.0/2 = 5.0$$

In a computer you can only retain a certain number of significant digits for each number. This means that whenever a result cannot be exactly represented -- like above, where we've added 5.01 and 5.03 in 3-figure arithmetic -- a round-off error is introduced. This shows again how even something as simple as the addition of numbers may require a little thought.

Blind Use of Formulas

All too often, blind use of a formula on a computer can lead to inaccurate answers. For example, if you blindly try to apply the formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

to determine the zeros of the quadratic $ax^2 + bx + c$,

you will lose figures of accuracy quite often. In this example, if you are working strictly to three figures throughout, you will obtain the answers x = -2 or 0 when you apply the formula to $x^2 + 2x + 0.001$.

A better technique avoids the cancellation (giving the 0) and uses the fact that the product of the roots is 0.001. Thus the smaller answer may be more accurately calculated as 0.001/(-2) or –0.0005.

From a strictly mathematical point of view, there are two equivalent formulas for calculating variance of a set of numbers about their mean, viz.

$$s_n = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$s_n = \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left[\sum_{i=1}^{n} x_i\right]^2\right)$$

However, in the second formula, if you are using eight figure arithmetic, for example, you cannot accurately represent the squares of many numbers. For X^T = (10000, 10001, 10002), using 8 figure arithmetic, the results are 1.0 for the first formula but 0.0 using the second formula.

An ill-conditioned problem will, by definition lead you astray, but its converse may also help speed results. For example, consider

$I_r = e^{-1} \; {}_0^1 \; e^x x^r dx$, for r = 0, 1, 2,…,20

You could routines for evaluating this integral by numerical quadrature, to whatever accuracy is required, for each value of r. But a very much faster method would use the recurrence relation in the backwards direction, viz.

$I_{r-1} = 1 - I_r / r$, with $I_N = 0$, where N (> 20) depends on the accuracy required but is also determinable by simple and very rapid numerical experiment, or in this simple case, by elementary analysis. However, don't make the mistake of using the more obvious forward recurrence that fails to produce accurate results beyond the first few values of r with only single-precision arithmetic.

Thus,

$$I_r = 1 - rI_{r-1}, \text{ with } I_0 = 1 - e^{-1}$$

examples a very ill-conditioned problem.

<u>Pen and Paper Methods Do Not Always Work</u>

Pen and paper methods do not always work in a computer. One class of these problems are those that are grossly inefficient if real world considerations require numbers that are too large for practical use of a computer.

Consider for example, Cramer's rule ($x_r = D_r/D$, where D_r is the determinant of A with the r^{th} column replaced by b and D is the determinant of A. In theory you can find each individual element of x, but it is extremely inefficient.

<u>Summary --- Errors Are Inherent to Computer Arithmetic</u>

If the data of your problem are exact, and if the problem has a unique solution, then you would expect results to be accurate to a specified number of figures. But, as the above examples show, in computer arithmetic, the ease of getting a correct answer, for example with single precision arithmetic, will depend on the sensitivity of the results to perturbations in the data. The first problem is that in a computer even the representation of numbers cannot usually be performed exactly. Second, there are inevitable computer rounding off errors in most of the meaningful computations that are germane to portfolio management issues. How important these errors are depends on the stability of the numerical method chosen. To the extent that the problem is ill-conditioned, even small perturbations from inexact representation or round-off will create large changes in the answer.

David Sayers is a Senior Technical Consultant and John Morrissey is a Computational Mathematician, of the Numerical Algorithms Group (www.nag.com). They can be reached via email at infodesk@nag.com.

Notes

1. To arm financial managers dissecting errors in their results, the Numerical Algorithms Groups (NAG) provides detailed discussions of how machine errors can impact complex financial computations and similar calculations in its online library documentation (see www.nag.com, especially Chapter X02 of the NAG Library).