The poor quality of random numbers

After decades in which the army of computer simulators have used machine-generated random numbers with confidence, even aplomb, some generating algorithms turn out to be frayed at the edges.

WHAT exactly is a random number? Many senses in which that question may be understood evidently make no sense. For example, it is nonsensical to ask in the abstract whether "3" is a random number. For one thing, it cannot be described as random without specifying the range of numbers from which it has been chosen. For another, even if it has been chosen by some random process (from among, say, the integers between "1" and "10" inclusive), that will be entirely irrelevant unless the uses to which the number is put are specified.

In this spirit, one might set out to calculate the scattering cross-section of a moving billiard ball by another at rest, building a simple computer program to simulate the typical collision and then assigning trajectories chosen at random to consecutive impactors. Then one becomes an important user of random numbers. The same result can be obtained more easily with a little calculus, but not all calculations are that amenable to analytical treatment. In any case, what matters in such a context is that the numbers should be random with respect to each other. By now, there are many procedures, known by the generic name of Monte Carlo, that follow these lines and which are major users of the commodity called the random number, which has to be manufactured somehow.

That is not child's play. Here is an account (by Robert M. Ziff, *Phys. Rev. Lett.* **69**, 2671; 1992) of one technique:

> "...a total of 6×10^{12} random numbers were generated, requiring a few months computing time on about a dozen computer workstations (Apollo 425, IBM R/S 6000) running simultaneously."

Random numbers made in ways like this are not really random, but are the products of computational schemes or algorithms. So, in the last resort, a set of random numbers cannot be entirely free from non-randomness. Everybody in the trade knows that, and tries to arrange that calculations are so designed that the non-randomness will not embarrassingly become apparent. But that seems not always to be possible.

Indeed, for the past two years there has been a zephyr (hardly a gale) of excitement that a calculation by Alan M. Ferrenberg, D. P. Landau and Y. Joanna Wong, from the University of Georgia at Athens, appears to have produced erroneous results in circumstances that cast doubt on the quality of the random numbers they used (*Phys. Rev. Lett.* **69**, 3382–3384; 1992). This was at the time more than a mere cloud on a blue horizon; some of the random-number generators used go back for twenty years, and had not previously been found wanting.

The test system used by Ferrenberg and his colleagues was the familiar Ising lattice, a two-dimensional square lattice supposed occupied by classical spins pointing in one direction or the opposite. The Ising lattice is a model for several kinds of critical phenomena. The usual procedure is to simulate an infinitely large lattice by supposing it to be broken into finite rectangles and imposing the same boundary conditions at the edges of each rectangle making up the whole plane.

The object of the exercise is to calculate the thermodynamic properties of such a system, which is a matter of calculating the energy of particular configurations of spins and, from them, the partition function (a function of the temperature). The snag is that, except when the lattice is small, the enumeration problem is huge. So people have fallen back on Monte Carlo methods to arrive at the equilibrium state of the system in one go. The simple way is to invert a single spin, chosen at random, and to calculate the effect on the thermodynamic properties of flipping the spins of its nearest neighbours, *their* nearest neighbours, and so on. That state in which a single flipped spin makes no overall difference is the equilibrium state.

The practical difficulty is that the simulation methods become extremely slow near the critical temperature at which the lattice is transformed from the ordered to the disordered state. So Ferrenberg and his colleagues were anxious to try a novel method due to Ulli Wolff from the University of Kiel, in which the approach to equilibrium in an Ising lattice involves switching whole clusters of spins simultaneously. This was reckoned to be much faster, but Ferrenberg found that it gave "systematically incorrect results". They knew that because the simple two-dimensional Ising lattice is exactly calculable.

Three Finnish workers have now shown that the errors found by Ferrenberg and colleagues are indeed traceable to the random numbers used (Vattulainen, I., Ala-Nissila, T. & Kankaala, K. *Phys. Rev. Lett.* **73**, 2513–2516; 1994). They test their conclusions with a variety of generators of random numbers. An account of the manufacture of random numbers follows.

Everything is done by computer, so that numbers are most conveniently represented as binary digits, preferably as many as there are bits in one computer byte. Evidently an iterative process, in which one random number is generated from others already defined, is simplest when there is a need to make a lot of them. Suppose that the numbers each consist of 32 binary bits ("0"s and "1"s) and that there are already n of them in being, say $x_1, x_2 ... x_n$. Then one could imagine forming x_{n+1} by using some simple computer operation to combine some specified pair of the pre-existing numbers together. One of the simplest computer operations is the "OR" or "XOR" operator which, operating one bit at a time, turns one 32-bit number into another. That is the basis of what is called the generalized feedback shift register (GFSR) generator of random numbers, in which $x_n = x_{n-p} \otimes x_{n-q}$, where $\otimes$ means the OR operator and p and q are integers.

There are more elaborate ways of making random numbers, but to the extent that they all depend on algorithms, they are all subject to non-randomness. But it is clear that the GFSR method requires at least p or q numbers (whichever is the larger), whence the benefits of the algorithm $x_n = (16807 \times x_{n-1})\mathrm{mod}(2^{31}-1)$, apparently variously known as GGL, CONG and RAN3.

The Finnish group has repeated the simulations that caused a stir at the end of 1992, discovering that discrepancies do indeed depend on the random-number generator. They have gone on to devise two neat physical tests of the randomness of random numbers. One, called the "random walker test", would have the random numbers make a point move between one of the four quadrants in a square, in the expectation that it would occupy each of the four sub-squares equally often. In reality, they find that some favourite algorithms, especially the GFSR generators with the larger of p and q equal to 31, 250 and 521, as well as RAN3, all fail the simple test.

The other test is even simpler. Suppose the random numbers are scaled to fit between 0 and 1, and that they are taken together in blocks of n as they come off the battery of perpetually running workstations. Take the average of each block and assign the number 0 if the average is less than 0.5 and, otherwise, 1. There should be an equal number of the two digits, and the probability of departures from equality can be obtained by simple statistics. Again the same random-number generators fail.

Primarily, all this will be a warning to the Monte Carlo community, but others will be surprised that the errors have come to light so late in the day, and that so little should be known of their origin. Perhaps that is something with which to occupy the long Finnish winter nights ahead. **John Maddox**