Monte Carlo at Work

by Gary D. Doolen and John Hendricks

Every second nearly 10,000,000,000 "random" numbers are being generated on computers around the world for Monte Carlo solutions to problems that Stan Ulam first dreamed of solving forty years ago. A major industry now exists that has spawned hundreds of full-time careers invested in the fine art of generating Monte Carlo solutions—a livelihood that often consists of extracting an answer out of a noisy background. Here we focus on two of the extensively used Monte Carlo solvers: MCNP, an internationally used neutron and photon transport code maintained at Los Alamos; and the "Metropolis" method, a popular and efficient procedure for computing equilibrium properties of solids, liquids, gases, and plasmas.

MCNP

In the fifties, shortly after the work on the Monte Carlo method by Ulam, von Neumann, Fermi, Metropolis, Richtmyer, and others, a series of Monte Carlo transport codes began emerging from Los Alamos. The concepts on which these codes were based were those outlined by von Neumann (see "Stan Ulam, John von Neumann, and the Monte Carlo Method"), but a great deal of detailed work was needed to incorporate the appropriate physics and to develop shorter routes to statistically valid solutions.

From the beginning the neutron transport codes used a general treatment of the geometry, but successive versions added such features as cross-section libraries, variance-reduction techniques (essentially clever ways to bias the random numbers so that the guesses will cluster around the correct solution), and a free-gas model treating thermalization of the energetic fission neutrons. Also, various photon transport codes were developed that dealt with photon energies from as low as 1 kilo-electron-volt to the high energies of gamma rays. Then, in 1973, the neutron transport and the photon transport codes were merged into one. In 1977 the first version of MCNP appeared in which photon cross sections were added to account for production of gamma rays by neutron interactions. Since then the code has been distributed to over two hundred institutions worldwide.*

The Monte Carlo techniques and data now in the MCNP code represent over three hundred person-years of effort and have been used to calculate many tens of thousands of practical problems by scientists throughout the world. The types of problems include the design of nuclear reactors and nuclear safeguard systems, criticality analyses, oil well logging, health-physics problems, determinations of radiological doses, spacecraft radiation modeling, and radiation damage studies. Research on magnetic fusion has used MCNP heavily.

The MCNP code features a general three-dimensional geometry, continuous energy or multigroup physics packages, and sophisticated variance reduction techniques. Even very complex geometry and particle transport can be modeled almost exactly. In fact, the complexity of the geometry that can be represented is limited only by the dedication of the user.

---

*The MCNP code and manual can be obtained from the Radiation Shielding Information Center (RSIC), P.O. Box X, Oak Ridge, TN 37831.

PAIR-DISTRIBUTION FUNCTION

This plot gives the probability of pairs of charged particles in a plasma being separated by a certain distance. The probabilities are plotted as a function of the distance between the pair of particles (increasing from left to right) and temperature (decreasing from front to back). At the left edge, both the distance and the probability are zero; at the right edge, the probability has become constant in value. Red indicates probabilities below this constant value, yellow and green above. As the temperature of the plasma decreases, lattice-like peaks begin to form in the pair-distribution function. The probabilities, generated with the Metropolis method described in the text, have been used for precise tests of many theoretical approximations for plasma models.

The Metropolis Method

The problem of finding the energy and configuration of the lowest energy state of a system of many particles is conceptually simple. One calculates the energy of the system, randomly moves each particle a small distance, and recalculates the energy. If the energy has decreased, the new position is accepted, and the procedure continues until the energy no longer changes.

The question of how to calculate equilibrium properties of a finite system at a given temperature is more difficult, but it was answered in a 1953 *Journal of Chemical Physics* article by Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller, who decided that the calculation should follow the same steps for finding the minimum energy but with one important change. When a move results in an *increased* energy, one accepts the new position with probability $e^{-\Delta E/T}$, where ΔE is the change in energy and T is the temperature. This procedure gives the equilibrium solution for any physical system. In fact, a system with many particles can be solved with only a few lines of code and a fast computer.

Although calculations for short-range forces are much easier than for long-range forces (such as the Coulomb force), the Metropolis technique has been used for most physical systems in which the forces between particles are known. Wayne Slattery, Hugh DeWitt, and one of the authors (GD) applied the technique to a neutral Coulomb plasma consisting of thousands of particles in a periodic box. The purpose was to calculate such physical properties as the temperature at which this type of plasma freezes and the pair distribution function, which is the probability of finding one particle at a given distance from another (see accompanying figure). Because the uncertainty in a Monte Carlo result is proportional to $1/\sqrt{N}$, where N is the number of moves of a single particle, several million moves requiring several hundred Cray hours were needed to obtain accurate results for the plasma at many temperatures.

As computers become faster and their memories increase, larger and more complicated systems are being calculated far more accurately than even Stan Ulam probably expected. ■