BNL-34641

MONTE CARLO SIMULATION OF THE MICROCANONICAL ENSEMBLE

Michael Creutz

Department of Physics
Brookhaven National Laboratory
Upton, New York 11973

Talk given at the Lattice Gauge Theory Workshop, Argone National Lab., April 5 - 7, 1984

MONTE CARLO SIMULATION OF THE MICROCANONICAL ENSEMBLE

Michael Creutz
Brookhaven National Laboratory, Upton, NY 11973
Physics Department

ABSTRACT

We consider simulating statistical systems with a random walk on a constant energy surface. This combines features of deterministic molecular dynamics techniques and conventional Monte Carlo simulations. For discrete systems the method can be programmed to run an order of magnitude faster than other approaches. It does not require high quality random numbers and may also be useful for nonequilibrium studies.

In this talk I will discuss a new algorithm for simulating statistical systems[1]. It combines features of canonical Monte Carlo procedures and microcanonical molecular dynamics calculations. The method has some advantages over other methods. First when applied to discrete systems such as the Ising model, it can give rise to extremely fast programs, as much as an order of magnitude faster than other techniques. Second, it does not require high quality random numbers. Indeed, for Ising systems no external random numbers are needed at all; the system can generate its own. Third, the algorithm can be set up so that all energy flow must occur through the system itself. This results in an energy conserving dynamics which may be a useful model for nonequilibrium phenomena. Fourth, the three sets of beautiful Monte Carlo renormalization group results presented at this conference made me realize that my algorithm can serve to determine coupling constants on blocked lattices.

This is, after all, a conference on lattice gauge theory, and my algorithm can easily be applied to these systems. For continuous gauge groups, it brings a lattice into equilibrium in a time competitive with, but not particularly better than, a good conventional Monte Carlo program. The reason we gain little here is that most computer time in a gauge theory simulation is spent multiplying neighboring links to calculate the action. The new algorithm cannot avoid this arithmetic. My unfounded hope is that by delving into alternative simulation schemes, one may gain insight into new ways to treat anticommuting integrals numerically.

Let me begin by reminding you of the two conventional approaches to the numerical simulation of statistical systems. First is the canonical Monte Carlo method as exemplified in the algorithm of Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller[2]. Here one sets up a Markov chain of configurations

$$C_1 \to C_2 \ldots C_i \to C_{i+1} \ldots \quad (1)$$

Each configuration in this sequence is obtained stochastically with a probability distribution which depends only on the previous configuration. Thus the algorithm is defined by the probability matrix $M(C',C)$ for taking configuration C into configuration C'. This matrix is constructed in such a way that the ultimate probability density for encountering any given configuration in the sequence is given by the Boltzmann weight

$$P(C) \propto \exp(-\beta H(C)) \quad (2)$$

where for particle physics $H(C)$ is the action and for condensed matter physics it is the energy of configuration C. Thus the Markov process generates a canonical ensemble of configurations. In this class of algorithms, I also include the differential versions as exemplified by the stochastic Langevin equation[3].

A molecular dynamics simulation, on the other hand, begins with storing the coordinates and momenta of a dynamical system, for example, a set of atoms. Introducing a force law between these particles, one then numerically integrates Newton's equations of motion. This sets up a deterministic meandering on the surface of constant total energy, mimicking the evolution of a real system. Note that such an algorithm uses no random numbers. Unless there is some interesting non-ergodicity, the complexity of the system will generate its own randomness. Note also that the procedure makes no reference to the parameter β. Here the temperature is an output, not an input. It is usually determined from the equipartition of energy, which says that the average kinetic energy per degree of freedom is $kT/2$. Callaway and Rahman[4] have introduced this method to lattice gauge theory by adding a canonical momentum conjugate to each link variable and setting up a Hamiltonian dynamics in the fifth dimension of computer time. Polonyi and Wyld[5] have recently advocated using this scheme for fermionic simulations, as discussed by Polonyi at this conference.

The microcanonical Monte Carlo method combines features of both these techniques. Here I use a Markov chain of configurations, as in the canonical approach. However, instead of using the Boltzmann factor to weight changes, I constrain the energy of the system to be constant. Thus the algorithm consists of a random walk on the surface of constant energy. In general this surface is a rather complicated submanifold of the space of all configurations. For certain models, such as the O(3) σ-model or SU(2) lattice gauge theory, a heat bath Monte Carlo algorithm can be easily modified to implement such a constraint[6]. However, for a general system, it is convenient to introduce an auxiliary variable, referred to as a demon, to transfer energy around the lattice. This demon carries an energy E_d. To keep the demon from arbitrarily scrambling the lattice, E_d must be constrained.

Although later I will discuss further constraints, for now I just require the demon energy to be positive. Including the demon, the microcanonical "partition function" under study is

$$Z_{mc} = \sum_{E_d>0} \sum_{C} \delta\big(H(C) + E_d - E_0\big) \tag{3}$$

where E_0 is the action of the surface under consideration. To simulate this system numerically, we sweep over the lattice variables along an arbitrary path, as in a conventional Monte Carlo program. When the demon visits some variable, he attempts to change it. For example, in a gauge theory the demon will tentatively multiply a link variable by a group element from a table such as would be used in a Metropolis program. If the new configuration has lower action, the demon makes the change and puts the change of action into his sack

$$E_d \to E_d + H(C) - H(C') \tag{4}$$

where C and C' are the old and new configurations, respectively. If the new configuration has higher action, the demon makes the change if and only if he has enough energy, that is if the right hand side of (4) is positive. Otherwise he leaves the lattice and his energy unchanged and proceeds to try another change of the lattice. As in the usual Metropolis procedure, the demon may try a fixed number of "hits" on a given variable before moving on to another; this is often advantageous in gauge theories where considerable arithmetic on the neighbors of a link does not have to be repeated for each hit.

As with molecular dynamics simulations, the temperature is not an input parameter. The randomness of the system is determined by the initial action E_0. The temperature can easily be determined from the average value of the demon energy. For a large system, standard statistical mechanical arguments relate the microcanonical sum in (3) to the canonical partition function

$$Z = \sum_{E_d>0} \sum_{C} \exp\big(-\beta(H(C) + E_d)\big) \ . \tag{5}$$

The demon thus decouples from the system and we have

$$\beta = 1/\langle E_d \rangle \ . \tag{6}$$

This equation will receive straightforward modifications if the spectrum of demon energies is discrete or if additional constraints are imposed. We remark in passing that an amusing variational exercise shows that this particular estimate of the temperature gives the least variance. This is true even for a generalized demon with an arbitrary density of energy states.

This microcanonical algorithm clearly has close connections with the Metropolis et al. technique. Indeed, it is trivial to take a program using the latter method and convert it to the former. To go the other direction, one merely has to let the demon visit a heat bath between updatings. That is, if before visiting some variable the demon energy is replaced with a new value chosen randomly with a weighting given by the Boltzmann factor

$$P(E_d) \propto \exp(\beta E_d) \quad , \tag{7}$$

then the algorithm reduces precisely to that of Ref. (2). This observation has an important consequence for demons making large jumps on a large lattice. In this case, before visiting any given site the demon will have been wandering on essentially uncorrelated distant parts of the lattice. As far as the local region of the new site is concerned, the demon has used the distant lattice as a heat bath and his distribution will appear as in Eq. (7). Thus we reach the crucial conclusion that locally the algorithm can be no worse, or no better than the Metropolis et al procedure in terms of number of sweeps required to decorrelate lattice configurations. The microcanonical advantage for discrete systems is that those sweeps can take much less computer time, for reasons I will discuss later.

In ordinary Monte Carlo one picks a value of β and runs the computer to calculate correlation functions, such as the average action or energy. In the microcanonical approach one picks the action and runs to find the temperature. It is possible to interpolate between these extremes by introducing a large number of demons. A single demon can only carry a small amount of energy compared to the lattice and thus the lattice energy is essentially fixed. However, many demons can collectively hold an appreciable energy. If their number is comparable to the number of degrees of freedom of the lattice one effectively studies a curve in the energy versus temperature plane

$$\langle H\rangle + \alpha/\langle\beta\rangle = \text{const.} \tag{8}$$

where the parameter α depends on the relative number of demons and lattice variables. When α is small the simulation becomes microcanonical, and when it is large we go over into a canonical distribution.

In conventional Monte Carlo discussions a condition of detailed balance is usually used to justify an algorithm. It is natural to ask if a similar condition can be applied to microcanonical algorithms. On a surface of constant energy the usual detailed balance condition reduces to the statement that a change from configuration C to C' should have the same probability as would a change from C' to C. Actually this is a stronger condition that necessary. In any stochastic simulation one only needs that the desired equilibrium distribution, i.e. Boltzmann distribution for the canonical ensemble or an equidistribution on an energy surface for microcanonical, be stable under the algorithm. This allows some rather bizarre moves. For example, in the Ising model simulation to be discussed later in this talk, I have 60 demons represented by two bit numbers. After each lattice sweep I shuffle the first bits and then independently the second bits. This will in general give an entirely new set of demon energies, but the operation is justified because if all demon energy states are equally likely, they still will be after the transformation.

Let me now turn to a brief discussion of finite volume effects in the microcanonical ensemble[7]. It is, of course, only in the infinite volume limit that the canonical and microcanonical distributions give the same thermodynamics. There are two effects here that I would like to discuss. First, consider the demon energy distribution. When the lattice is large it effectively is a heat bath interacting with the demon and thus the demon's energy should be exponentially distributed as in Eq. (7). However, if the lattice is small it is not a perfect heat bath. Indeed, when the demon has a lot of energy, the lattice has less and is thus somewhat cooler. The demon's distribution can be more properly obtained from the differential equation

$$(d/dE_d)\, \ell n(P(E_d)) = -\,\beta(E_{latt} = E_T - E_d)\ . \tag{9}$$

Here E_T is the total energy of the demon-lattice system and E_{latt} is the portion of that energy in the lattice. The function $\beta(E)$ is the inverse temperature corresponding to a lattice with energy E. Expanding the right-hand side of (9) about E_T and solving the resulting equation gives

$$P(E_d) \propto \exp(-\,\beta E_d - (\beta E_d)^2/(2CV)) + O(1/V^2))\ . \tag{10}$$

Here V is the volume of the lattice and C its specific heat

$$C = -\,\beta^2/V\ \ \partial E_{latt}/\partial\beta\ . \tag{11}$$

In practice the V^{-1} correction in (10) is quite small and goes to zero at critical points where C diverges. A similar correction to the kinetic energy distribution applies to conventional molecular dynamics calculations.

Another finite volume phenomenon appears on comparing a general correlation function in the canonical and microcanonical ensembles. Let Γ be any correlation function of interest. Its expectation in the microcanonical ensemble is

$$\Gamma_M(E) = \frac{\sum\limits_C \delta(H(C) - EV)\Gamma(C)}{\sum\limits_C \delta(H(C) - EV)} \tag{12}$$

where $\Gamma(C)$ is the value of the correlation in configuration C. Here the demons are implicitly included and I have made the volume explicit by letting E by the energy density on the lattice. We wish to compare this quantity with its value in the canonical ensemble, which can be obtained from a Boltzmann distribution of microcanonical ensembles

$$\Gamma_C(\beta) = \frac{\int dE\ \rho(E,V)\Gamma_M(E)\ e^{-\beta EV}}{\int dE\ \rho(E,V)\ e^{-\beta EV}} \ . \tag{13}$$

Here $\rho(E,V)$ represents the density of states for the system at energy density E and volume V. Now take this equation and expand Γ_M around $\langle E\rangle$, the canonical expectation of E at inverse temperature β. This gives

$$\Gamma_C(\beta) = \Gamma_M(E) + \frac{1}{2}\langle E - \langle E\rangle^2\rangle_C \frac{\partial^2}{\partial E^2}\Gamma_M(E) + \ldots . \tag{14}$$

The second term in this relation can be related to the specific heat via the Einstein-Gibbs[8] fluctuation dissipation theorem. Thus microcanonical and canonical correlation functions differ by the finite volume correction

$$\Gamma_C(\beta) = \Gamma_M(\langle E\rangle) + \frac{C(\beta)}{2\beta^2 V}\left(\frac{\partial^2}{\partial E^2}\Gamma_M(E)\right) + O(V^{-2}) \ . \tag{15}$$

In Ref. (7) this correction was numerically observed for the two dimensional Ising model. Note that using E^2 for Γ reproduces the fluctuation dissipation theorem.

I have claimed that the microcanonical Monte Carlo procedure can use substantially less computer time than other methods for discrete systems. The reason for this is that the algorithm never needs any floating point operations and all arithmetic can be done with simple bit manipulations. Thus one can carry the multi-spin coding idea[9] to its ultimate. Even on a conventional serial machine, simultaneous processing of a number of spins equal to the length of the computer's words is possible. To see this explicitly, consider simulating the three dimensional Ising model on the CDC 7600, a computer which uses 60 bit words. On this machine I store 60 independent spins in each word. For a given x and y coordinate in the lattice, one word stores all spins with even z coordinate and another stores those with odd z. Thus the size of my lattice is forced to be 120 in the z direction, although one can easily generalize the program to any larger multiple of 60. To store a lattice of dimension I by J by 120 requires 2IJ words of memory. My demons, of which there are 60, reside in two further words, D1 holding the first bits of each demon and D2 the second bits. Each demon, then, is a two bit number and can take on values in the set $\{0,1,2,3\}$. All energy changes in an Ising model with an even number of nearest neighbors will be in units of 4. Removing this factor from the demon energy, I wish to hold constant

$$E_{latt} + 4\,E_d \tag{16}$$

where E_d denotes the total energy of the 60 demons. Being two bit numbers, the demons have both an upper and lower bound on their energies. From a general point of view this will still give the microcanonical ensemble, but one might worry that the extra constraint might excessively reject changes, thus slowing convergence. To see that this is not a serious problem, note that the extra rejection rate will depend on the relative Boltzmann weight for a demon to exceed 3. Including the factor of 4 above, we see that this effect is of order $\exp(-12\beta)$. In the three dimensional model the region of interest is $\beta \approx 0.22$ and thus the extra rejection is only a few percent.

As all quantities use every bit independently, arithmetic must be done "transversely". In other words all operations use only the Boolean unit, which is effectively 60 independent one bit processors. To illustrate the nature of the operations, consider some given spin. The tentative new demon energy associated with flipping this spin is

$$E'_d = E_d + (N_a - N_p)/2 = E_d + N_a - 3 \tag{17}$$

where N_a and N_p are respectively the number of antiparallel and parallel neighbors to the spin. The first step is to determine if a neighbor is antiparallel. If S and S1 denote words representing 60 pairs of neighboring spins, the exclusive or operation S xor S1 gives a word, call it W, with set bits wherever the respective spins are antiparallel. As indicated in Eq. (17) W should be added to the old demon energy. This can be done in a few logical instructions

```
B1 = D1 xor W
CARRY = D1 and W
B2 = CARRY  xor D2                                    (18)
REJECT = CARRY and D2
```

Here B1 and B2 are the first and second bits of the resulting sum and REJECT is a computer word with set bits indicating an overflow outside the allowed two bits for the demons. This set of steps is then repeated for all six spin neighbors and a few more operations remove the 3 in Eq. (17). After all of this is done, one can in one instruction simultaneously update the 60 spins in S

$$S \rightarrow S \text{ xor not REJECT} \qquad (19)$$

Similar operations accept or reject the changes in the demons.

Implementing these ideas on a CDC 7600, I update 29×10^6 spins per second (29 "megaflips"), or 24 megaflips while also accumulating the average demon energy and lattice magnetization. On a TI 99/4A I obtain 8 kiloflips on a 192×256 two dimensional model with video display but without measurements. For comparison[10], the fastest canonical programs on the CDC 7600 update at 2 to 3 megaflips, an order of magnitude slower. On other commercial machines, rates of 6 megaflips on the ICL DAP and 10 megaflips on the CDC Cyber 205 have been achieved. The Santa Barbara Ising processor, a special purpose machine designed for this problem, runs at 25 megaflips.

As an empirical demonstration that the scheme works, I will now show a few results on the 3 dimensional Ising model. In Fig. 1

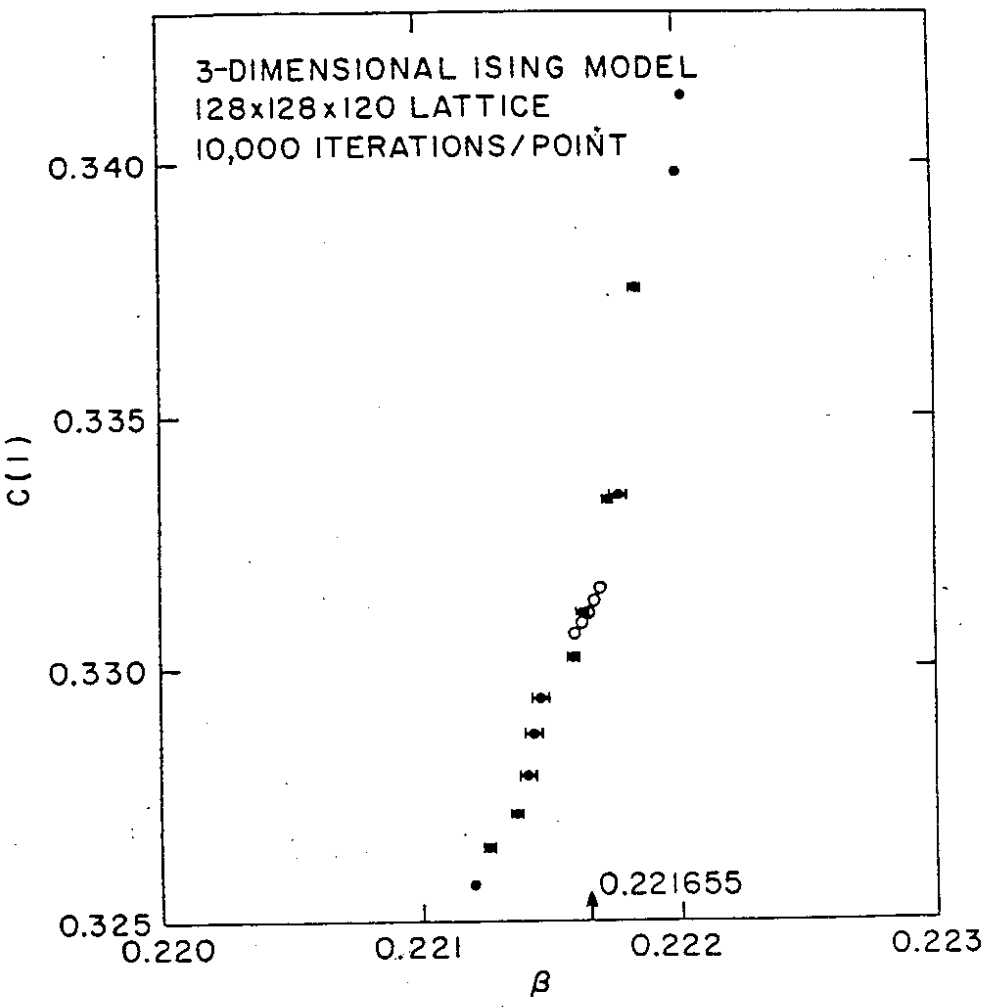

Fig. 1 The nearest neighbor correlation for the three dimensional Ising model plotted versus the inverse temperature in the critical region.

I show measurements of the inverse temperature as a function of the lattice energy in the critical region. These points represent an average over the last 9000 of 10000 sweeps over a 128 × 128 lattice The errors were obtained by averaging the data in 1000 iteration bins and assuming these were independent samples. On the graph are also four points from the Santa Barbara machine operating on a 64^3 lattice. Note both the rather expanded scale in the figure and the fact that the error bars run perpendicular to the direction they would in a standard Monte Carlo.

Fig. 2 shows an application of simple renormalization group ideas to the critical region of this model. If this critical behavior defines a field theory, then the correlation between spins should give a two point Greens function of the continuum theory. In general, however, there may be wave function renormalizations. These should cancel in the ratio of two correlations, giving a quantity which should remain finite in the continuum limit if the lengths are kept fixed in physical units. For more details in the context of a gauge theory application, see Ref. (1). The basic

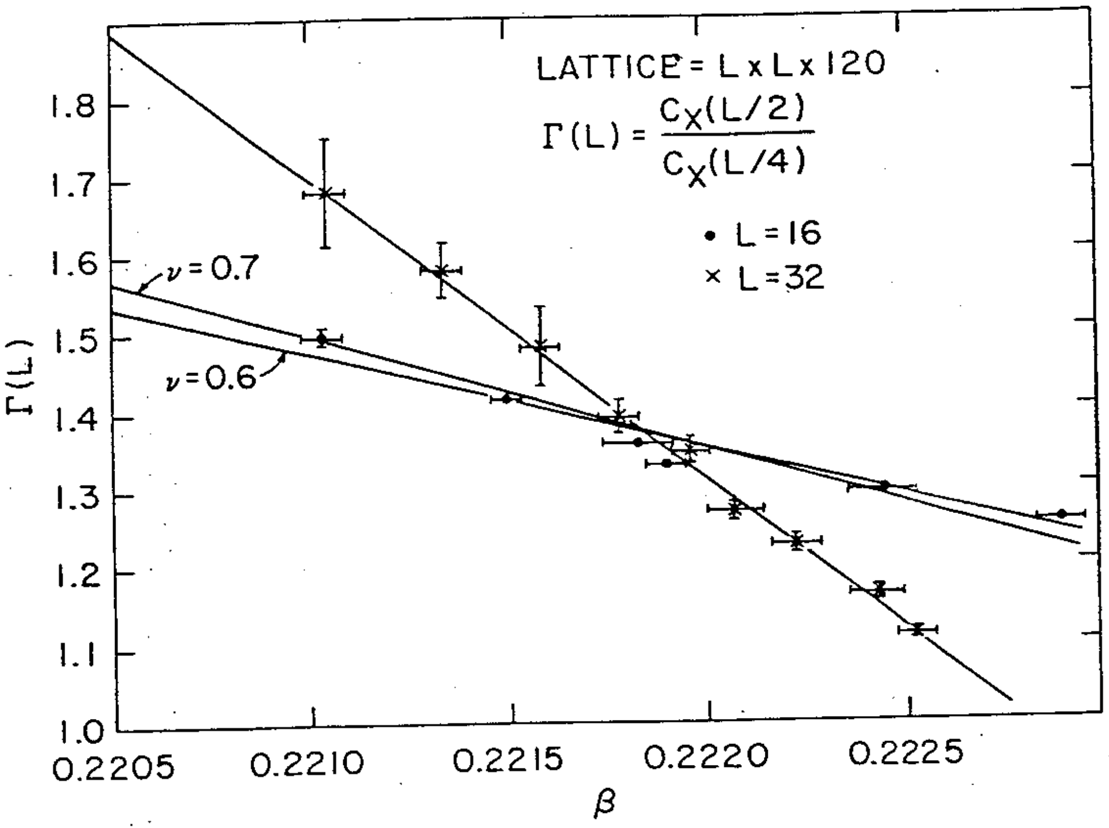

Fig. 2 The ratio C(L/2)/C(L/4) on a L × L ×120 lattice, where C(X) is the correlation between spins separated by X sites in the first coordinate. The crosses represent L = 32; the solid circles, L = 16.

conclusion is that at a critical point such a physical observable should be scale invariant, that is it should remain unchanged if we double all relevant lengths. In this figure we consider the ratio of the correlation of two spins half way around the lattice with two spins one quarter of the way around. This is done for two lattices with x × y being 16 × 16 and 32 × 32. The nature of the program makes it difficult to change the z dimension from 120, so I have kept the other scales much smaller. The correlations are also calculated in parallel using logic operations as discussed above. The critical point of the model shows up quite clearly as a crossing of the two sets of points. This crossover occurs slightly above the accepted critical beta of .221655, but this discrepancy may be due to further finite size effects. The relative slope of the two sets of points at the crossover is related to the exponent ν characterizing the divergence of the correlation length

$$\xi \approx |\beta - \beta_c|^{-\nu}$$

I have drawn an "eyeball" fit to the larger lattice data and then drawn the predicted slopes for the smaller lattice for $\nu = 0.6$ and 0.7. The data are clearly consistent with the accepted value $\nu = 0.64$.

Although this program is comparable in speed to the Santa Barbara processor, it cannot directly compete on this model. This is because a special purpose processor can be left on for very long runs without contention from other users. The real advantage of this new procedure is that it is easily generalized. Being written for a general purpose computer, it is straightforward to add other couplings. The mosts amusing way to do this is to add additional demons for each coupling constant. Thus one would generalize the microcanonical ensemble to place constraints not on just the total energy, but on each of the independent terms in the Hamiltonian. The various couplings would then be determined from the average value of the respective demons. This is potentially useful for Monte Carlo renormalization group studies, where one wishes to know the renormalized couplings on a blocked lattice.

As a slight variant on the microcanonical algorithm, one could let the 60 spins in any word come from 60 independent lattices. With shift instructions, the demons could transfer energy between these lattices. Thus one would construct the canonical ensemble from the microcanonical one in a textbook fashion, while simultaneously gaining 60 times the statistics of a naive non-multi spin-coded program.

Another amusing variant involves tying the demons to the sites and not have them move at all. Thus on each lattice site one would place 3 bits, one for the spin and two for its corresponding demon. Updating this system in a checkerboard fashion, i.e. first all even sites and then all odd ones, one would have a deterministic system of cellular automata simulating the Ising model. To conform with the strict definition of a cellular automaton, the checkerboard updating could be implemented with a fourth bit at each site, initially set to the site parity and flipping on each update. This bit would disable the demon when it is not his turn. The net result is a reversible Ising dynamics where energy transfer around the lattice only takes place through the lattice bonds. This may be useful for studies of nonequilibrium phenomena. Looked at from a distance, the lattice should simulate the heat equation, but without any floating point arithmetic; indeed, the computer would be doing what it likes best, bit manipulation.

I would like to draw one general conclusion from this talk. It is not that specifically this algorithm is a particularly better way of simulating some systems. Rather the point is that there can be orders of magnitude lurking in new ways of computing things. My hope is that even more orders of magnitude will be found in radically new techniques for simulating anticommuting fields, an area where current computer technology is inadequate for present algorithms.

REFERENCES

1. M. Creutz, Phys. Rev. Lett. 50, 1411 (1983).

2. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, J. Chem. Phys. 21, 1087 (1953).

3. G. Parisi and Y.-S. Wu, Sci. Sin. 14, 483 (1981).

4. D. Callaway and A. Rahman, Phys. Rev. Lett. 49, 613 (1982); Phys. Rev. D28, 1506 (1983).

5. J. Polonyi and H.W. Wyld, Phys. Rev. Lett. 51, 2257 (1983).

6. M. Creutz, in Proc. of IV Adriatic Meeting on Particle Physics, (1983) (in press).

7. G. Bhanot, M. Creutz and H. Neuberger, Nucl. Phys. B (in press)

8. J. Gibbs, Elementary Principles of Statistical Mechanics, Yale Univ. Press (1902); A. Einstein, Annalen der Physik 14, 354 (1904).

9. L. Jacobs and C. Rebbi, J. Comp. Phys. 41, 203 (1981).

10. D. Toussant, talk at Conference on Monte Carlo Methods and Future Computer Architectures, Brookhaven Lab. (1983).