18
Monte Carlo simulation I: the method

The lattice formulation reduces the Feynman path formula for the gauge theory into a multiple ordinary integral. This suggests that, at least for finite size systems, one might attempt to numerically evaluate the partition function. A moments thought, however, reveals that the high multidimensionality of the integrals makes conventional mesh techniques impractical. For example, consider a 10^4 site lattice, a size fairly typical for numerical work. Such a system has 40000 link variables. If we now take the simplest possible gauge theory, that with gauge group Z_2, the partition function becomes an ordinary sum. But this sum has an enormous number of terms, that number being

$$2^{40\,000} = 1.58 \times 10^{12\,041}. \tag{18.1}$$

Even if we could add one term in the time it takes light to pass by a proton and continue for the age of the universe, we would not put a perceptible dent in the sum.

The appearance of such large numbers immediately suggests a statistical treatment. Indeed, there are also an enormous number of ways to place molecules of H_2O into a glass and yet one only needs a few to determine the thermodynamic properties of water. The goal of the Monte Carlo approach is to provide a small number of configurations which are typical of thermal equilibrium in the statistical analog. Whereas the super-astronomical number of terms indicated in eq. (18.1) can never be summed exactly, it is straightforward to store the few tens of thousands of numbers characterizing typical configurations which strongly dominate the sum.

A Monte Carlo program begins with some initial configuration of the fields stored as an array in a computer memory. It then sequentially makes pseudo-random changes on these variables in such a manner that the ultimate probability density for encountering any configuration C is proportional to the famous Boltzmann factor

$$p_{\text{eq}}(C) \propto \mathrm{e}^{-\beta S(C)}, \tag{18.2}$$

where $S(C)$ is the action associated with the given configuration. In this chapter, to emphasize the connection with statistical mechanics, we

https://doi.org/10.1017/9781009290395.019 Published online by Cambridge University Press

explicitly display the factor of β which we previously absorbed in the definition of the action. Our goal is to use the computer as a 'heat bath' at inverse temperature β.

The Monte Carlo simulation technique is quite old in statistical physics (Metropolis *et al.*, 1953). It provides the possibility of performing 'experiments' on virtual 'crystals' with interactions governed by an arbitrary Hamiltonian of choice. This in principle enables isolation of various dynamical features and their role in such phenomena as phase transitions. Furthermore, as the entire configuration is stored, any desired correlation function is in principle available. The technique converges well in both high and low temperature regimes and interpolates nicely in between. This latter point is of particular import to the particle physicist, who desires to relate the Wilson demonstration of confinement in strong coupling to the continuum field theory obtained in the weak coupling limit.

As with real experiments, Monte Carlo simulations have certain inherent sources of error. Statistical fluctuations are always present, and only decrease with the square root of the computing time. This can be a serious handicap if one is interested in some detailed parameter displaying fluctuations comparable to the signal. Then one must run a hundred times longer to merely reduce the errors to 10%. In addition, systematic effects may arise from the finite lattice size and spacing. For the four-dimensional systems considered here, the linear size of the lattice is necessarily quite limited, eight to ten sites on a side being typical. (At this writing, the largest lattice yet studied for a gauge theory had 16^4 sites; Bhanot and Rebbi, 1981.) Finally, a systematic error arises in determining when equilibrium has been reached; in particular, one must worry about being trapped in some metastable state.

Many of these systematic effects are readily amenable to further study. The lattice size is easily varied over a limited range and indeed observation of finite size effects can provide useful information on the states of the theory (Brower, Creutz and Nauenberg, 1982). Different initial conditions can test for thermal equilibrium; some possible starting states will be discussed later. Finite lattice spacing effects are of interest because they are tied to the renormalization of the bare coupling, as extensively discussed in chapters 12 and 13.

Regarding the computer as merely a heat bath immediately suggests the most intuitive Monte Carlo algorithm (Yang, 1963). We successively touch this heat bath to all the links in the lattice. A real thermal source in contact with a link would cause that variable to fluctuate thermally throughout the group manifold. When the source is removed, the link would be left

in any of its allowed states with a probability given by the associated Boltzmann weight. For example, to mimic this process for the gauge group $Z_2 = \{1, -1\}$, one would begin by calculating the probability of the given link to be left in the state $+1$

$$P(1) = e^{-\beta S(1)}/(e^{-\beta S(1)} + e^{-\beta S(-1)}). \tag{18.3}$$

Here $S(\pm 1)$ is the action evaluated with the link in question in the corresponding state and all other links held fixed at their current values. Note that if the action is local, that is if only nearby links are directly coupled, then this probability depends solely on these nearby links. The algorithm continues by asking the computer for a randomly selected number from a uniform distribution between zero and one. If $P(1)$ exceeds this number, the link is set to unity, otherwise it is set to -1. The entire procedure is then repeated on the next link and so forth until the entire lattice is covered. This represents one Monte Carlo iteration and generates the next state in a Markov chain of configurations.

These ideas are applicable to any group. The 'heat bath' algorithm replaces each group element with a new value selected randomly with a weighting given by the current exponentiated action. When applied to an ensemble of states, this gives a new ensemble which is closer to an equilibrium ensemble in a sense that we will shortly make precise.

When the group manifold is intricate, the above selection procedure for new group elements may be impractical or too time consuming to implement efficiently. For this reason computationally simpler algorithms are often used. These are also constructed to bring one closer to equilibrium, but may require more iterations to achieve the same convergence. If each iteration takes less computer time than a heat bath selection, there can be a net gain.

To design alternative procedures, we need a criterion for determining when an algorithm for randomly changing an ensemble of configurations will tend towards equilibrium. In general, each state in the Monte Carlo sequence results from a Markovian process applied to the previous configuration. Each stage in the algorithm is thus specified by a probability distribution $P(C', C)$ for taking configuration C into C'. An obvious necessary condition on P is that it leave an equilibrium ensemble in equilibrium. Thus the Boltzmann weights should be an eigenvector of P

$$e^{-\beta S(C)} = \sum_{C'} P(C, C') e^{-\beta S(C')}. \tag{18.4}$$

Remarkably, if the algorithm has eventual access to any configuration, this is also a sufficient condition for any ensemble to ultimately approach the Boltzmann distribution of eq. (18.2).

To demonstrate this claim, we need a notion of 'distance' between ensembles. Suppose we have two ensembles E and E', each of many configurations. Denote the probability density for configuration C in E or E' by $p(C)$ or $p'(C)$, respectively. Then we define the distance between E and E' as the sum

$$\|E-E'\| = \sum_C |p(C)-p'(C)|, \tag{18.5}$$

where the sum is over all possible configurations. Now suppose that E' resulted from the application of a Monte Carlo algorithm satisfying eq. (18.4) to ensemble E. This means that

$$p'(C) = \sum_{C'} P(C, C')p(C'). \tag{18.6}$$

As $P(C', C)$ is a probability, it satisfies

$$P(C', C) \geqslant 0, \tag{18.7}$$

$$\sum_{C'} P(C', C) = 1. \tag{18.8}$$

We can now compare the distance of E' from equilibrium with the distance of E from equilibrium

$$\begin{aligned}\|E'-E_{\text{eq}}\| &= \sum_C |\sum_{C'} P(C, C')(p(C')-p_{\text{eq}}(C'))| \\ &\leqslant \sum_{C, C'} P(C, C')|p(C')-p_{\text{eq}}(C')| = \|E-E_{\text{eq}}\|.\end{aligned} \tag{18.9}$$

We conclude that the algorithm reduces the distance of an ensemble from equilibrium. Note that if $P(C, C')$ never vanishes, this inequality is strict unless we are already in equilibrium.

To insure that an algorithm has the equilibrium distribution as an eigenvector, most algorithms in practice are based on products of steps each satisfying a condition of detailed balance

$$P(C', C)\,\mathrm{e}^{-\beta S(C)} = P(C, C')\,\mathrm{e}^{-\beta S(C')}. \tag{18.10}$$

Summing over the second index C' and using eq. (18.8) immediately gives the eigenvector eq. (18.4).

The detailed balance condition, which is sufficient but not necessary for the approach to equilibrium, far from uniquely specifies the matrix $P(C, C')$. The heat bath algorithm automatically satisfies the condition because $P(C, C')$ is independent of C' and proportional to the Boltzmann weight for C. Metropolis *et al.* (1953) used the detailed balance criterion to formulate another algorithm which, because of calculational simplicity, has become the most popular in practice. For the gauge theory, we begin with the selection of a trial U' as a tentative replacement for some link variable U. The test variable is selected with a distribution $P_T(U, U')$ which

https://doi.org/10.1017/9781009290395.019 Published online by Cambridge University Press

must be symmetric in U and U'

$$P_T(U, U') = P_T(U', U). \tag{18.11}$$

Beyond this constraint, P_T is arbitrary and may be selected empirically to optimize convergence. Normally it is best if U' has a weighting towards the old value of U. Once U' is chosen, we evaluate the tentative new action $S(U')$ for comparison with its old value $S(U)$. If the action is lowered, that is if the new configuration has a larger Boltzmann weight, then this change is accepted. The detailed balance condition then determines the remainder of the algorithm: if the action is raised the change must be accepted with conditional probability $\exp(-\beta(S(U')-S(U)))$.

A simple way to implement this procedure is to obtain U' by multiplying U with a random group element from a table, where this table is itself of random elements with a convenient weighting towards the identity. The table should contain enough elements to generate for the group and should contain the inverse of each of its elements in order to satisfy eq. (18.11).

The Metropolis algorithm described above has an essential dependence on two parameters. The first is the weighting of the random changes towards the identity. This peaking should become more severe at low temperatures where large changes would be routinely rejected. A popular criterion for selecting this distribution is to make the acceptance probability at any step roughly 50%.

A second parameter in the algorithm is the number of trial changes attempted on any given link before going on to the next. In most statistical problems this is taken to be one; however, for the gauge theory the interaction is rather complicated and requires considerable arithmetic to evaluate. This means that it can be extremely beneficial to do as good a job as possible in selecting the stochastic changes. In terms of real computer time involved in reaching equilibrium, it is usually of value to test ten or more new elements, during which time the multiplication of neighboring elements appearing in the action need not be repeated. Note that as the number of tries, or 'hits' increases, the Metropolis algorithm approaches the heat bath. This is because repeating the procedure on one link will ultimately bring that link into thermal equilibrium with its temporarily fixed neighbors. This is what the heat bath does in one step. To determine an optimum number of hits, one can simply make a few trial runs on a small lattice to study the convergence in real time.

Although the Metropolis procedure brings an ensemble closer to equilibrium by less per iteration than the heat bath, it has the advantage of being extremely simple. The detailed form of the group measure is not

needed; the algorithm automatically generates it with a random walk around the group. Furthermore, to change the form of the action or group is straightforward. Nevertheless, for groups with simple enough manifolds the heat bath algorithm may be rather elegantly implemented.

To illustrate some techniques for generating variables with a given weight, we will now discuss the heat bath generation of $SU(2)$ elements for the gauge theory with the Wilson action (Creutz, 1980*b*). First we need a source of random numbers uniformly distributed between zero and one. Such generators are standard in most high level computer languages, and we assume a good one has been provided (Knuth, 1969). The important point for our purposes is that calls to such a generator are extremely fast, comparable to a multiplication, and thus usually represent only a minor part of the time of a simulation.

Given a source of random numbers with such a uniform distribution, we can easily produce a random sequence with an arbitrary distribution. Suppose we have some positive function $f(x)$ on the unit interval and wish to generate points with a weighting proportional to f. For simplicity assume that f is bounded; if not, make a change of variables to make it so. Without loss of generality, we assume that $f(x)$ is bounded by unity. Using the given random number generator, we obtain a trial number for the first element of our weighted sequence. Calling this number x, we obtain a second random number and accept x if the new random variable is less than $f(x)$. This is repeated many times to form a sequence of accepted values of x. As the probability of accepting any x is proportional to $f(x)$, the sequence has the desired weighting.

This process will be inefficient if the function f is strongly peaked. In this case we may need to generate many points before one is accepted. If one knows approximately where the peak is and its form, one may be able to change variables to spread it out. This forms the basis for the following $SU(2)$ algorithm.

While working on a particular link (ij), we need consider only the contribution to the action coming from the six plaquettes containing that link. If we denote by $\tilde{U}_\alpha$, $\alpha = 1, \ldots, 6$, the six products of three link variables which interact with the link in question, then the probability distribution for the new link variable under the heat bath algorithm is

$$\mathrm{d}p(U) \sim \mathrm{d}U \exp\left(\tfrac{1}{2}\beta \operatorname{Tr}\left(U \sum_{\alpha=1}^{6} \tilde{U}_\alpha\right)\right). \tag{18.12}$$

For $SU(2)$ the trace is automatically real. In chapter 8 we parametrized $SU(2)$ as the surface of a four-dimensional sphere

$$SU(2) = \{a_0 + \mathrm{i}\mathbf{a}\cdot\boldsymbol{\sigma} \mid a_0^2 + \mathbf{a}^2 = 1\}, \tag{18.13}$$

https://doi.org/10.1017/9781009290395.019 Published online by Cambridge University Press

where σ represents the Pauli matrices. The invariant group measure is uniform over this sphere

$$dU \sim d^4a\,\delta(a^2-1). \tag{18.14}$$

This representation shows the useful property that a sum over any number of $SU(2)$ elements is proportional to another element of the group. In particular, we have

$$\sum_{\alpha=1}^{6} \tilde{U}_\alpha = k\bar{U}, \tag{18.15}$$

where $\bar{U}$ is an element of $SU(2)$ and k is the determinant

$$k = \left|\sum_{\alpha=1}^{6} \tilde{U}_\alpha\right|^{\frac{1}{2}}. \tag{18.16}$$

The utility of this observation appears when we use the invariance of the group measure to absorb $\bar{U}$

$$dp(U\bar{U}^{-1}) \sim dU \exp(\tfrac{1}{2}\beta k \operatorname{Tr} U) \sim d^4a\,\delta(a^2-1)\exp(\beta k a_0). \tag{18.17}$$

Thus we have found the peak in the exponentiated action and rotated it to the identity. We have reduced the problem to generating points randomly on the surface of the unit sphere in four dimensions with an exponential weighting along the a_0 direction. Given an element U generated in this manner, we replace the link variable on the lattice with the product

$$U'_{ij} = U\bar{U}^{-1}. \tag{18.18}$$

To generate the appropriately weighted points on the sphere, we first do the integration over the magnitude of $\mathbf{a}$ with the delta function and obtain

$$dU \exp(\tfrac{1}{2}\beta k \operatorname{Tr} U) \sim \tfrac{1}{2} da_0 d\Omega\,(1-a_0^2)^{\frac{1}{2}} \exp(\beta k a_0). \tag{18.19}$$

Here $d\Omega$ is the differential solid angle for the vector $\mathbf{a}$, which has length $(1-a_0^2)^{\frac{1}{2}}$. We need the generate a_0 in the interval $(-1, 1)$ with probability density

$$dp(a_0) \sim (1-a_0^2)^{\frac{1}{2}} \exp(\beta k a_0)\, da_0 \tag{18.20}$$

and the direction for $\mathbf{a}$ is totally random. For moderate to large β, the dominant peaking in eq. (18.20) comes from the exponential factor. This can be removed with a change of variables from a_0 to

$$z = \exp(\beta k a_0). \tag{18.21}$$

Equation (18.20) now becomes

$$dp(z) = dz\,(1-\beta^{-2}k^{-2}\log^2 z)^{\frac{1}{2}}. \tag{18.22}$$

The generation of z can proceed as outlined earlier; with the random number generator a trial z is selected randomly in the allowed interval

$$e^{-2\beta k} \leqslant z \leqslant e^{+2\beta k} \tag{18.23}$$

and this is rejected with the probability given on the right hand side of eq. (18.22). Repeating this until a z is accepted, one reconstructs a_0 by

https://doi.org/10.1017/9781009290395.019 Published online by Cambridge University Press

taking a logarithm. The final step in the algorithm is to randomly select the direction for **a**. This can be done in a variety of ways; for example, one could generate a random point in the interior of a three-dimensional sphere and use its direction from the origin. Note that in the above discussion several tricks special to the group $SU(2)$ were used. To find corresponding tricks for a new group or even a new action can be tedious and thus most simulations in practice have turned to the Metropolis algorithm.

Monte Carlo computer programs tend to be physically rather short and straightforward. They begin with a set of nested loops over the various links. The selection of the new variables, such as outlined above, involves only a few rather simple operations. The multiple loops result in these steps being repeated on the order of a million times. The $SU(2)$ procedure is readily implementable so that it requires less than 200 microseconds per link on a *CDC* 7600 computer. The group $SU(3)$ with a reasonably optimized algorithm uses one to two milliseconds per link on the same machine. In both these cases, the majority of the time is spent multiplying the neighboring group elements. In practice it is usually computer time rather than storage which limits these programs. For $SU(2)$ it is convenient to store the four components of a_μ for each link, resulting in a relatively modest 160000 numbers for a 10^4 site lattice.

We now turn to describe some simple Monte Carlo 'experiments'. An obvious first question involves the time required to reach equilibrium. When we are not operating near a phase transition this time can be remarkably short. In figure 18.1 we show the results of several runs with the heat bath algorithm on the group $SU(2)$. The coupling constant was set to the constant value

$$\beta = 4g_0^{-2} = 2.3 \tag{18.24}$$

which was selected as representative of the slowest convergence occurring with this model. Runs are shown on four-dimensional lattices of from 4^4 to 10^4 sites. Each iteration represents one application of the heat bath algorithm to every lattice link; on the 10^4 lattice one such step represents 40000 new $SU(2)$ elements. As a function of the number of iterations, we plot the average plaquette or expectation of the action

$$P = \langle 1 - \tfrac{1}{2} \operatorname{Tr} U_\square \rangle, \tag{18.25}$$

discussed in chapter 9. For each size lattice, two different initial configurations were studied. The + symbols represent an ordered start, with all link matrices set to the identity. This ground state of the statistical system corresponds to beginning at zero temperature. In contrast, the crosses represent an initial configuration where each element was selected randomly, uniformly in the invariant measure, from the entire group. In this

case we start at infinite temperature. Thus we approach equilibrium from opposite extremes. Note that in all cases convergence appears to be essentially complete after only 20 to 30 iterations. Thermal fluctuations, which must always be present, are quite apparent on the small lattices but become relatively small on the 10^4 site system.

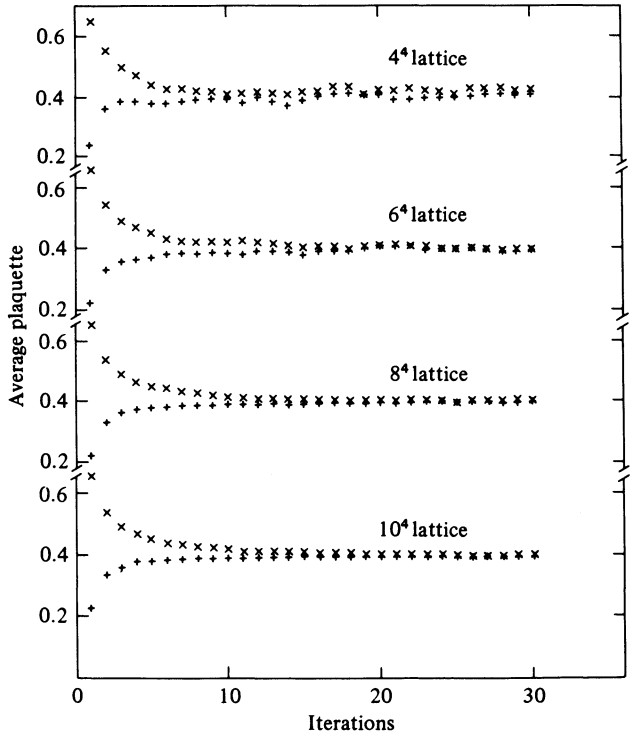

Fig. 18.1. Several Monte Carlo runs with the group $SU(2)$ (Creutz, 1980*b*).

The situation can be considerably less advantageous if a phase transition is nearby. In figure 18.2 we show the convergence of the $U(1)$ theory on a 6^4 lattice at $\beta = 1.0$. On an infinite lattice this system exhibits a second-order phase transition at $\beta = 1.012 \pm .005$ (Lautrup and Nauenberg, 1980*a*; DeGrand and Toussaint, 1980; Bhanot, 1981). In addition to the rather slow convergence when compared to the $SU(2)$ case, note the large fluctuations, characteristic of critical behavior.

The above runs illustrate the two simplest initial conditions, corresponding to zero and infinite temperature. In the case of a first-order transition such initial states can result in the lattice being caught in a metastable state. As in a real experiment, a random (ordered) lattice can be supercooled

(superheated) substantially below (above) the transition temperature without settling in a reasonable time into the correct phase. To aid in approaching equilibrium one can add a 'seed' consisting of an ordered (disordered) piece of the lattice. This motivates a third interesting initial configuration consisting of a lattice which is half ordered and half disordered. For example, links from sites with fourth coordinate less than half the lattice size could be randomized and the remainder ordered. In

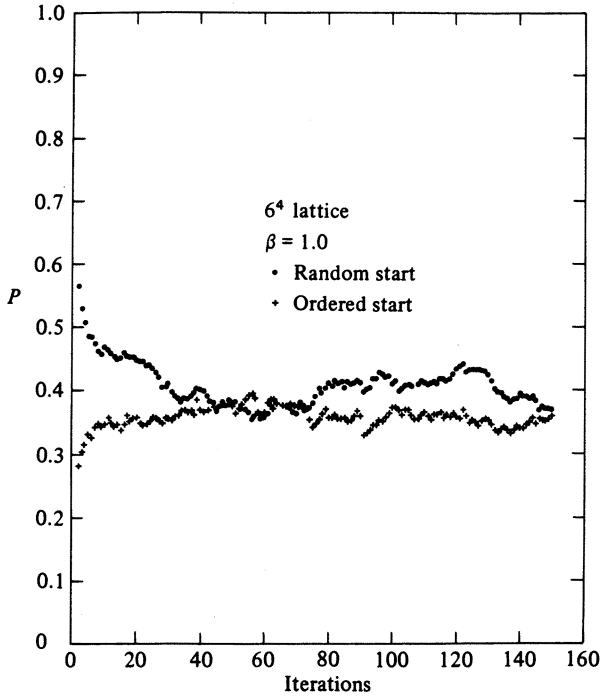

Fig. 18.2. Two Monte Carlo runs with the $U(1)$ model near its critical point.

figure 18.3 we show several Metropolis algorithm runs with such a start for the gauge group Z_2 on an 8^3 by 20 lattice (Creutz, Jacobs, and Rebbi, 1979*b*). Several values of temperature are selected near the transition point as calculated in chapter 16. The figure shows a linear drift characteristic of one phase 'dissolving' the other. The aimless drift very near the transition indicates that this method can rather accurately determine the temperature of the phase change. Indeed, this is analogous to putting some ice in water to determine its melting point.

The fact that convergence is fast away from phase transitions and slow near them suggests another type of experiment. Upon heating and then

https://doi.org/10.1017/9781009290395.019 Published online by Cambridge University Press

cooling the system through a range of temperatures, regions of slow convergence will appear as hysteresis effects. This provides a technique for rapidly locating regions of coupling for further study. In figure 18.4 we show the results of such thermal cycling on the $SU(2)$ model in four and five space-time dimensions and the $U(1) = SO(2)$ theory in four dimensions. Each point in this figure was obtained by running on the order of twenty iterations with the heat bath algorithm from either a hotter or cooler

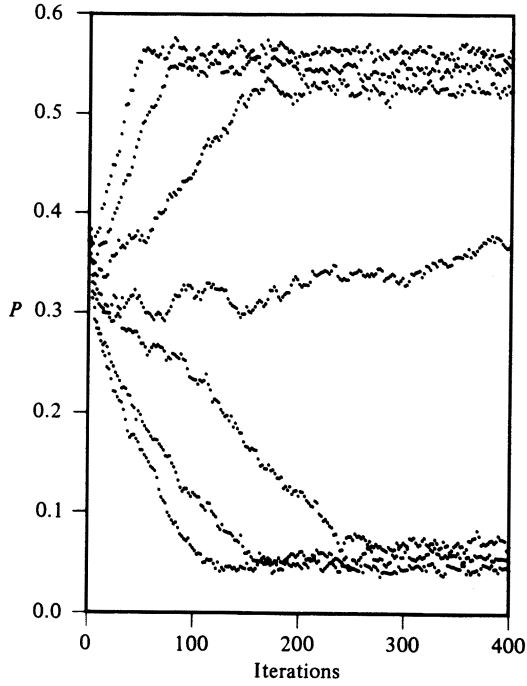

Fig. 18.3. Several Monte Carlo runs for the Z_2 model with mixed phase initial conditions. From the top down, these runs take β from 0.41 to 0.47 in steps of 0.01 (Creutz, Jacobs and Rebbi, 1979*a*).

configuration. As a check on normalizations, we also plot the lowest order strong and weak coupling results. Phase transitions are to be suspected in those regions where the heating and cooling cycles do not agree, as clearly observed for the five-dimensional $SU(2)$ and four-dimensional $U(1)$ models. Further analysis in the transition region suggests that the $U(1)$ transition is second order (Lautrup and Nauenberg, 1980*a*) and the five-dimensional $SU(2)$ transition is first order. As the latter fits the prediction of mean field theory, we conclude that $d = 5$ is close to $d = \infty$.

This is further supported by the fact that the $U(1)$ model in five dimensions also exhibits a first-order transition (Bhanot and Creutz, 1980).

The hysteresis seen in figure 18.4*c* may at first seem a bit surprising because the transition is believed to be second order and should have a continuous internal energy regarded as a function of the temperature.

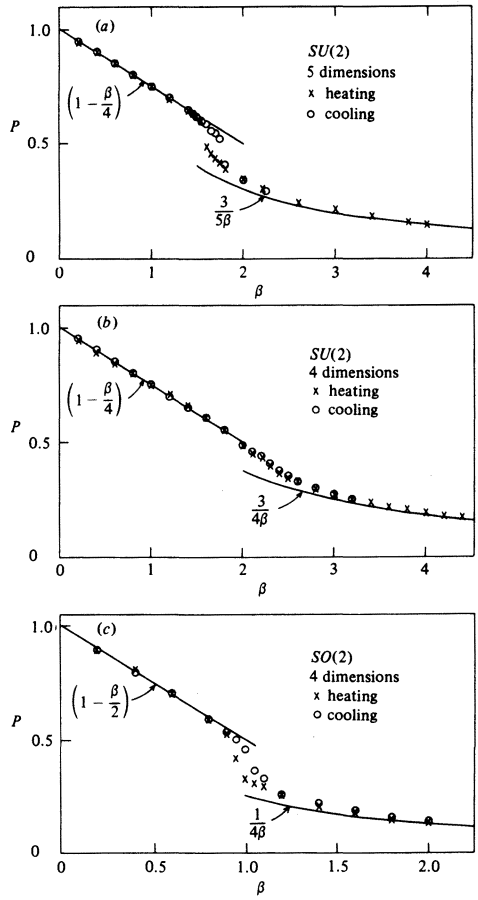

Fig. 18.4. Thermal cycles on several of the models (Creutz, 1979).

However, this thermal cycle was rather rapid, and, as figure 18.2 shows, a few tens of iterations are insufficient for relaxation of the energy near the critical point. Repeating this type of experiment at different cycle rates does indeed provide information on the nature of a transition. With a strong first order phase change, superheating and supercooling result in a hysteresis cycle which is reasonably stable in shape and relatively inde-

pendent of the speed with which the system is heated or cooled. The closing of the cycle is predominantly determined by the temperatures at which metastable minima of the free energy disappear. In contrast, the cycle associated with a second-order transition tends to close continuously as the experiment becomes more adiabatic.

The four-dimensional $SU(2)$ model exhibits a sharp contrast to the other systems in figure 18.4. It displays no clear structure other than a fairly rapid crossover from strong to weak coupling behavior at β around two. Careful analysis of the specific heat in this region shows a definite peak but no signal of a real singularity (Lautrup and Nauenberg, 1980*b*). This result supports the desired absence of a phase transition in this non-Abelian system. In figure 11.1 we showed the internal energy of the $SU(3)$ model as obtained from Monte Carlo analysis. It exhibits a rapid crossover qualitatively similar to the $SU(2)$ case.

Problem

1. On your home computer, write a Monte Carlo program to simulate the one-dimensional Ising model. Calculate the internal energy as a function of temperature and compare with the exact result.
2. Devise a heat bath algorithm for the gauge group $U(1)$.

https://doi.org/10.1017/9781009290395.019 Published online by Cambridge University Press