Lattice gauge notes IV
Monte Carlo

Start with the pure gauge theory; return to the quarks later.

The path integral

$$Z = \int (dU) e^{-S}$$

on a finite lattice is a finite dimensional integral. One might try to evaluate it numerically. But it is a many dimensional integral. With $SU(3)$ on 10^4 lattice we have $4 * 8^4$ links, each parametrized by 8 numbers. Thus it is a $320,000$ dimensional integral. Taking two sample points for each direction, this already gives

$$2^{320,000} = 3.8 \times 10^{96,329} \qquad \text{terms}$$

The age of the universe is only $\sim 10^{27}$ nanoseconds, so adding one term at a time will take a while.

Such big numbers suggest a statistical approach. Find "typical" equilibrium configurations

$$P(C) \sim e^{-\beta S(C)}$$

Use a Markov process

$$C \to C' \to \ldots$$

For example, consider a Z_2 theory with $U \in \{\pm 1\}$. Then the path integral is a discrete sum over $2^{40,000}$ terms. But storing a configuration involves only $40,000$ numbers. Given a configuration, set up a loop over those numbers, touch a heat bath to each in turn. That is, calculate the probability for that spin to be 1

$$P(1) = \frac{e^{-\beta S(1)}}{e^{-\beta S(1)} + e^{-\beta S(-1)}}$$

and then set it to 1 or -1 with this probability.

For a 146 line (with comments) example program, see

"http://thy.phy.bnl.gov/~creutz/coursenotes/z2/".

For simple models, such as $SU(2)$, this procedure can be implemented fairly straightforwardly. For $SU(3)$ the generation of the heat bath weighted group elements is complicated so other algorithms are used.

In general we want a procedure that takes a configuration C to a new one with some probability $P(C \to C')$. This satisfies $0 \leq P \leq 1$ and $\sum_{C'} P(C \to C') = 1$. (For

continuous groups the sum really means integrals.) To set up a Markov process, P should have no dependence on the history (an easy way to make a mistake!).

We want this to bring us closer to "equilibrium." For this we need at least two things.

- equilibrium should be an "eigen-distribution"

$$\sum_{C'} P(C \to C') e^{-S(C')} = e^{-S(C)}$$

- ergodicity: all states can eventually be reached with the set of P's used

This is all that is required!

Suppose we start with ensembles of states, E, characterized by their distribution $p(C)$. A distance between ensembles is easily defined

$$D(E, E') \equiv \sum_C |p(C) - p'(C)|$$

This is positive and vanishes only if the ensembles are equivalent. A step of our Markov process takes ensemble E into another E' with

$$p'(C) = \sum_{C'} P(C \to C') p(C')$$

Now assume that P is chosen so that the equilibrium distribution $p_{eq}(C) = e^{-S(C)}/Z$ is an eigenvector of eigenvalue 1. Compare the new distance from equilibrium with the old

$$D(E', E_{eq}) = \sum_C |p'(C) - p_e q| = \sum_C |\sum_{C'} P(C \to C')(p(C) - p_{eq}(C))|$$

Now the absolute value of a sum is always less than the sum of the absolute values, so we have

$$D(E', E_{eq}) \leq \sum_C \sum_{C'} P(C \to C') |(p(C) - p_{eq}(C))|$$

The since each C must go somewhere, the sum over C' gives unity and we have

$$D(E', E_{eq}) \leq \sum_C |(p(C) - p_{eq}(C))| = D(E, E_{eq})$$

Thus the algorithm automatically brings one closer to equilibrium.

How can one be sure that equilibrium is an eigen-ensemble? The usual way is detailed balance, a sufficient but not necessary condition. This states that the forward and backward rates between two states are equal when in equilibrium

$$p_{eq}(C) P(C \to C') = p_{eq}(C') P(C' \to C)$$

Summing this over C' immediately gives the fact that the equilibrium distribution is an eigen-ensemble.

The heat bath algorithm trivially satisfies detailed balance since it uses $P(C \to C') \propto p_{eq}(C')$ with all untouched variables being fixed.

The Metropolis et al. approach is an elegant and simple way to construct an algorithm satisfying detailed balance. This begins with a trial change on the configuration, specified by a trial probability $P_T(C \to C')$. This is required to be constructed in a symmetric way, so that

$$P_T(C \to C') = P_T(C' \to C)$$

This by itself would just tend to randomize the system. To restore the detailed balance, the trial change is conditionally accepted with probability

$$A(C, C') = \min(1, p_{eq}(C')/p_{eq}(C))$$

In other words, if the Boltzmann weight gets larger, take the change, otherwise accept it with probability proportional to the ratio of the Boltzmann weights.

An explicit expression for the final transition probability is

$$P(C \to C') = P_T(C \to C')A(C, C') + \delta(C, C')\left(1 - \sum_{C''} P_T(C \to C'')A(C, C'')\right)$$

The delta function term accounts for the possibility that the change is rejected.

For lattice gauge theory with its U variables, the trial change can be most easily set up via a table of group elements $T = \{g_1, ...g_n\}$. The trial change consists of picking an element randomly from this table and using $U_T = gU$. These can be chosen arbitrarily with two conditions: (1) multiplying them together in various combinations should generate the whole group and (2) for each element in the table, its inverse must also be, i.e. $g \in T \Rightarrow g^{-1} \in T$. The second condition is essential for having the forward and reverse trial probabilities equal.

Note: the measure of the group is not used in any way! It is generated automatically.

Generally the group table should be weighted towards the identity. Otherwise the acceptance gets small and you never go anywhere. But not too much, because then the motion is slow. Usually the width of the table is adjusted to the acceptance is about 50%. For free field theory the optimum can be worked out, it is a bit less. Note: a big change with a small acceptance can sometimes be better than small changes; this appears to be the case with simulating self avoiding random walks.

Amusing observation: the acceptance involves $\frac{p_{eq}(C')}{p_{eq}(C)}$ The expectation of this ratio in equilibrium is

$$\left\langle \frac{p_{eq}(C')}{p_{eq}(C)} \right\rangle = \sum_C p_{eq}(C) \sum_{C'} P_T(C \to C') p_{eq}(C')/p_{eq}(C) == 1$$

since

$$\sum_C P_T(C \to C') = \sum_C P_T(C' \to C) = 1$$

Of course the average acceptance is not 1 since it is expectation of the minimum of this ratio and 1. I like to use this expectation as a monitor of when I am near equilibrium.

Variations:

1) calculation of the "staples" takes a fair amount of time, thus multiple hits are good

2) overrelaxation: the contribution to the action from a single link U is of the form $\mathrm{ReTr}(MU)$ with M depending only on the neighbors. Project M onto a group element U_0 in some arbitrary way. Then take $U_T = U_0^{-1} U^{-1} U_0$. This takes U to a reasonably far away place with little change in action. I.e., with constant M it takes the element to its inverse, which is on the opposite side of the minimum action.

3) Cabbibo Marinari: For $SU(2)$ a heat bath is not too hard to implement (my old paper). For $SU(3)$, apply this successively to three $SU(2)$ subgroups.

Old results:
- thermal cycles, measure plaquette
- Z_2: strong first order phase transition; area vs. perimiter clean
- $U(1)$: deconfining phase transition at $\beta = 1$. Blamed on monopole condensation. Order uncertain, weak first order?
- $SU(2)$ and $SU(3)$: smooth passage from weak to strong coupling. Argues for confinement.
- $SU(N)$, $N \geq 4$: bulk phase transition; lattice artifact?

Other studies:
- Higgs: confinement versus Higgs mechanism
- generalized actions: induce new artifacts

Fermions

The numerical difficulties with fermionic fields stem from their being anticommuting quantities. Thus it is not immediately straightforward to place them on a computer, which likes to manipulate numbers. Indeed, the Boltzmann factor is formally an operator in a Grassmann space, and cannot be directly interpreted as a probability for Monte Carlo purposes. All algorithms in current use eliminate the fermions at the outset by a formal analytic integration. This is possible because most actions in practice are, or can easily be made, quadratic in the fermionic fields. The fermion integrals are then over generalized gaussians. Unfortunately, the resulting expressions involve the determinant of a large, albeit sparse, matrix. This determinant introduces nonlocal couplings between the bosonic degrees of freedom, making the path integrals over the remaining fields rather time consuming. Nevertheless, various tricks have been developed to minimize the pain with fermions.

For this discussion I will be quite generic and assume we are interested in a path integral of form

$$Z = \int (dA)(d\psi)(d\psi^*) \ \exp(-S_G(A) - \psi^* M(A)\psi).$$

Here the gauge fields are formally denoted A and fermionic fields ψ and ψ^*. As I will be concentrating on fermionic details, I will ignore the technicality that the gauge fields are group elements. All details of the fermionic formulation are hidden in the matrix $M(A)$. While I may call A a gauge field, the algorithms are general, and have potential applications in other field theories and condensed matter physics.

Note that under multiplicative rescaling a Grassmann integral behaves as

$$\int d\psi f(\psi a) = (\int d\psi f(\psi))a.$$

This can be written in the heuristic form $d(\psi a) = (d\psi)/a$. For integration over several anticommuting variables, I have

$$\int (d\psi) f(M\psi) = |M| \int d\psi f(\psi)$$

where M is an arbitrary matrix, $|M|$ its determinant, and $(d\psi)$ denotes $d\psi_1 \ldots d\psi_n$. This immediately implies the Matthews-Salam formula for a fermionic Gaussian integral

$$\int (d\psi d\psi^*) \ e^{-\psi^* M \psi} = |M|$$

where $(d\psi d\psi^*) = d\psi_1 d\psi_1^* \ldots d\psi_n d\psi_n^*$.

Using this, I explicitly integrate out the fermions to convert the path integral to

$$Z = \int (dA) \ |M| \ e^{-S_G}.$$

This is now an integral over ordinary numbers and therefore in principle amenable to Monte Carlo attack.

For now I assume that the fermions have been formulated such that $|M|$ is positive and thus the integrand in can be regarded as proportional to a probability measure.
- Wilson fermions satisfy $\gamma_5 M \gamma_5 = M^\dagger$
- Implies $|M|$ real
- for large K it is not always positive If $|M|$ is not positive, one can always double the number of fermionic species, replacing M by $MM^\dagger$. The case where M is not positive is not yet well understood, and indeed in some such cases the path integral may not be well defined. These problems are probably closely connected with the difficulties of placing chiral fermions on the lattice. This issue is quite important; with a chemical potential present to give a background fermion density, the determinant is not in general positive. Indeed, it is not yet known how to simulate this physically important problem.

Direct Monte Carlo study of the partition function in this form is still not practical because of the large size of the matrix M. In our compact notation, this is a square matrix of dimension equal to the number of lattice sites times the number of Dirac components times the number of internal symmetry degrees of freedom. Thus, it is typically a tens of thousands by tens of thousands matrix, precluding any direct attempt to calculate its determinant. It is, however, generally an extremely sparse matrix because most popular actions do not directly couple distant sites. All the Monte Carlo algorithms used in practice for fermions make essential use of this fact.

Some time ago Weingarten and Petcher presented a simple "exact" algorithm. They observed that by introducing auxiliary set of complex scalar fields ϕ one can rewrite the path integral in the form

$$Z = \int (dA)(d\phi^* \; d\phi) \exp(-S_G - \phi^* M^{-1} \phi).$$

Thus a successful fermionic simulation would be possible if one could obtain configurations of fields ϕ and A with probability distribution

$$P(A, \phi) \propto \exp(-S_G - \phi^* M^{-1} \phi).$$

To proceed I will assume that M is a positive matrix so this distribution is well defined.

Note:
- if χ is gaussianly random
- $P(\chi) \sim e^{-\chi^2}$
- then $\phi = M\chi$ distributed as desired for two flavors
- $P(\phi) \sim e^{-(M^{-1}\phi)^2}$
- hard part is the A updating

Trick to generate gaussian random numbers:
- given uniform generator $dP/dz = const$

- call it twice, generate $r = \sqrt{-\log(z_1)}$ and $\theta = 2\pi z_2$
- $dP/dr \sim re^{-r^2}$
- $x = r\cos(\theta)$ and $y = r\sin(\theta)$ are distributed as desired
- $dxdy \sim rdrd\theta$
- $dP/dxdy \sim e^{-r^2} \sim e^{-x^2-y^2}$

While M^{-1} is the inverse of an enormous matrix, one really only needs $\phi^* M^{-1}\phi$, which is just one matrix element of this inverse. Furthermore, with a local fermionic action the matrix M is extremely sparse, the nonvanishing matrix elements only connecting nearby sites. In this case there exist quite efficient iterative schemes for finding the inverse of a large sparse matrix applied to a single vector. Thus it was proposed to directly simulate the partition function using a Gauss-Seidel algorithm to calculate $M^{-1}\phi$. Most recent work has turned to the conjugate gradient algorithm for this inversion.

The conjugate gradient algorithm has also been quite useful for fermionic studies in the "valence" or "quenched" approximation, where the Dirac equation is solved on gauge field configurations obtained by simulations ignoring the feedback of the dynamical quarks on the gauge fields themselves. This approximation, circumvents the problem of simulating dynamical quarks, but still imposes substantial computational demands when the lattices are large.

The conjugate gradient method to find $\xi = M^{-1}\phi$ works by finding the minimum over ξ of the function $|M\xi - \phi|^2$. The solution is iterative; starting with some ξ_0, a sequence of vectors is obtained by moving to the minimum of this function along successive directions d_i. The clever trick of the algorithm is to choose the d_i to be orthogonal in a sense defined by the matrix M itself; in particular $(Md_i, Md_j) = 0$ whenever $i \neq j$. This last condition serves to eliminate useless oscillations in undesirable directions, and guarantees convergence to the minimum in a number of steps equal to the dimension of the matrix. There are close connections between the conjugate gradient inversion procedure and the Lanczos algorithm for tridiagonalizing sparse matrices.

The procedure is a simple recursion. Select some arbitrary initial pair of non-vanishing vectors $g_0 = d_0$. For the inversion problem, convergence will be improved if these are a good guess to $M^{-1}\phi$. Then generate a sequence of further vectors by iterating

$$g_{i+1} = (Mg_i, Md_i)g_i - (g_i, g_i)M^\dagger Md_i$$
$$d_{i+1} = (Md_i, Md_i)g_{i+1} - (Md_, Mg_{i+1})d_i$$

This construction assures that g_i is orthogonal to g_{i+1} and $(Md_i, Md_{i+1}) = 0$. It should also be clear that the three sets of vectors $\{d_0, ...d_k\}$, $\{g_0, ...g_k\}$, and $\{d_0, ...(M^\dagger M)^k d_0\}$ all span the same space.

The remarkable core of the algorithm, easily proved by induction, is that the set of g_i are all mutually orthogonal, as are Md_i. For an N dimensional matrix, there can be no more than N independent orthogonal vectors. Thus, ignoring roundoff errors, the recursion in

Eq. (15) must terminate in N or less steps with the vectors g and d vanishing from then on. Furthermore, as the above sets of vectors all span the same space, in a basis defined by the g_i the matrix $M^\dagger M$ is in fact tri-diagonal, with (Mg_i, Mg_j) vanishing unless $i = j \pm 1$.

To solve $\phi = M\xi$ for ξ, simply expand in the d_i

$$\xi = \sum_i \alpha_i d_i.$$

The coefficients are immediately found from the orthogonality conditions

$$\alpha_i = (Md_i, \phi)/(Md_i, Md_i).$$

Note that if I start with the solution $d_0 = M^{-1}\phi$, then I have $\alpha_i = \delta_{i0}$.

This discussion applies for a general matrix M. If M is Hermitian, then one can work with better conditioned matrices by replacing the orthogonality condition for the d_i with (d_i, Md_j) vanishing for $i \neq j$.

In practice, at least when the correlation length is not too large, this procedure adequately converges in a number of iterations which does not grow severely with the lattice size. As each step involves vector sums with length proportional to the lattice volume, each conjugate gradient step takes a time which grows with the volume of the system. Thus the algorithm is expected to require computer time which grows as the square of the volume of the lattice. Such a severe growth has precluded use of this algorithm on any but the smallest lattices. Nevertheless, it does show the existence of an exact algorithm with considerably less computational complexity than would be required for a repeated direct evaluation of the determinant of the fermionic matrix.

- quenched approximation: use CG to find propagator
- fixed gauge field background
- construct meson propagators, extract masses
- ignores internal loops

Here and below when I discuss volume dependences, I ignore additional factors from critical slowing down when the correlation length is also allowed to grow with the lattice size. The assumption is that such factors are common for the local algorithms treated here. In addition, such slowing occurs in bosonic simulations, and I am primarily concerned here with the extra problems presented by the fermions.

One could imagine making trial changes of all lattice variables simultaneously, and then accepting or rejecting the entire new configuration using the exact action. The problem with this approach is that a global random change in the gauge fields will generally increase the action by an amount proportional to the lattice volume, and thus the final acceptance

rate will fall exponentially with the volume. The acceptance rate could in principle be increased by decreasing the step size of the trial changes, but then the step size would have to decrease with the volume. Exploration of a reasonable region of phase space would thus require a number of steps growing as the lattice volume. The net result is again an exact algorithm which requires computer time growing as volume squared.

- step size ϵ
- $P_A \sim e^{-\epsilon^2 V}$
- pick $\epsilon \sim 1/\sqrt{(V)}$
- random walk in N steps goes $\epsilon\sqrt{N}$
- new configuration takes V steps
- order V^2 algorithm

So far this discussion has assumed that the trial changes are made in a random manner. If, however, one can properly bias these variations, it might be possible to reduce the volume squared behavior. The "hybrid Monte Carlo" scheme does this with a global accept/reject step on the entire lattice after a microcanonical trajectory. This algorithms has an interesting theoretical volume dependence that I now discuss.

In some sense, the difficulties with fermions stem from the time consuming evaluation of $M^{-1}\phi$. For simplicity, let me not write ϕ explicitly and assume that I have some action which is particularly difficult to calculate; so, I want to evaluate it as rarely as possible. Make the trial changing table into dynamical variables. Consider one group element g corresponding to each U. Look for a coupled distribution

$$P(g,U) \sim e^{-S(U)+\frac{1}{\epsilon^2}\mathrm{ReTr}g}$$

For the trial change of U and g take

$$\begin{aligned} g_{1/2} &= F(U)gF(U) \\ U' &= g_{1/2}U \\ g' &= F(U')g_{1/2}F(U') \end{aligned}$$

where for now $F(U)$ is an arbitrary "force" function taken to lie in the gauge group. Note that $dUdg = dU'dg'$. Also note that if we take g' to its inverse and repeat the same steps, we will go exactly back to the start.

$$\begin{aligned} g_{1/2}^{-1} &= F(U')g^{-1}F(U') \\ U &= g_{1/2}^{-1}U \\ g^{-1} &= F(U)g^{-1}F(U) \end{aligned}$$

Now in equilibrium both g and its inverse are equally likely, so a technically valid Monte Carlo step is to invert the g's. The above change followed by such an inversion and a Metropolis acceptance will give a valid algorithm.

The value added by introducing the driving F lies in trying to cancel the changes in the total action to get a bigger acceptance. To see this more easily, let me drop the group complications and replace (g, U) with simple numbers (p, A) and look for

$$P(p, A) = e^{-H(p,A)}$$

with

$$H = p^2/2 + S(A)$$

Define $F = \partial S/\partial A$. An approximately energy conserving algorithm is given by a discretization of Newtons law

$$\begin{aligned} p_{1/2} &= p + \delta F(A)/2 \\ A' &= A + \delta p \\ p' &= p_{1/2} + \delta F(A')/2 \end{aligned}$$

or combined

$$\begin{aligned} A' &= A + p\delta + F(A)\delta^2 \\ p' &= -p - (F(A) + F(A'))/2\delta. \end{aligned}$$

This is an area preserving map of the (A, p) plane onto itself. The scheme iterates this mapping several times before making the accept/reject decision. This iterated map remains reversible and area preserving. The second order terms in this equation make it equivalent to the leap frog procedure with an initial half step. The procedure thus generates a microcanonical trajectory.

The important point is that after each step the momentum remains exactly the negative of that which would be required to reverse the entire trajectory and return to the initial variables. If at some point on the trajectory I were to reverse all the momenta, the system would exactly reverse itself and return throughtr the same set of states from whence it came. Thus a final acceptance with the appropriate probability still makes the overall procedure exact. This slight modification of the hybrid algorithm makes it exact, just as the procedure with a single step removes the systematic errors of Langevin evolution. After each accept/reject step, the momenta p are refreshed, their values being replaced by new Gaussian random numbers. The fields ϕ could also be refreshed at this time, or less often, as turns out to be appropriate. The goal of the procedure is to use the microcanonical evolution as a way to restrict changes in the action so that the final acceptance will remain high for reasonable step sizes.

This procedure contains several parameters which can be adjusted for optimization. First is N_{mic}, the number of microcanonical iterations taken before the global accept/reject step and refreshing of the momenta p. Then there is the step size δ, which presumably should be set to give a reasonable acceptance. Finally, one can also vary the frequency with which the auxiliary scalar fields ϕ are updated.

The goal of the approach is to speed flow through phase space by replacing a random walk of the A field with a coherent motion in the dynamaical direction determined by the

conjugate momenta. As long as the total microcanonical time for a trajectory is smaller than some characteristic time for the system, the net change in A will grow linearly with both N_{mic} and δ; thus we have

$$\Delta A \sim N_{mic}\delta.$$

which should be valid as long as

$$N_{mic}\delta < \mathcal{O}(1).$$

With large N_{mic}, the change in the classical energy will also grow. In any given microcanonical step the energy changes by an amount of order δ^3. For N_{mic} of order δ^{-1}, the total energy change will then be of order δ^2. Because the evolution preserves areas in phase space, I have for the expected energy change

$$\langle H' - H \rangle = \frac{1}{2}\langle (H' - H)^2 \rangle + \mathcal{O}((H' - H)^3) = \mathcal{O}(\delta^4).$$

Now if I update V independent variables together, these positive contributions can coherently add and earlier arguments give an overall acceptance falling as

$$P_{\rm acc} \sim \exp(-CV\delta^4).$$

This means that δ should be taken to decrease with volume as $V^{-1/4}$. Correspondingly, N_{mic} should grow as $V^{1/4}$. The final result is that the total time required to obtain a substantially changed lattice grows as

$$T \sim V^{5/4}$$

This may be only an asymptotic statement, valid for systems much larger than the correlation length. The main uncertainty lies in the unknown characteristic time scales.

Experience has shown that the simple leapfrog algorithm is quite adequate in most practical cases. One should bear in mind, however, in going to large systems that higher order schemes may eventually become useful.

This method does, however, require the fermion matrix to be a square, requiring at least two species for Wilson fermions and eight for the Kogut Susskind case. Users of the hybrid algorithm without the global accept-reject step have argued for adjusting the number of fermion species by inserting a factor proportional to the number of flavors in front of the pseudofermionic term when the gauge fields are updated. This modification is simple to make, but is not completely theoretically understood. Indeed, it may introduce spurious behavior if the physics is not smooth in the number of flavors. Nevertheless, this flexability of the hybrid algorithm has made it quite popular, although when the global accept condition can be applied, it is probably worthwhile for the extra confidence it supplies.

Despite the successes of these fermion algorithms, the overall procedure remains somewhat awkward, particularly when compared with the ease of setting up a pure bosonic simulation.

This appears to be due to the non-local actions resulting from integrating out the fermions. Indeed, had one integrated out a set of bosons coupled quadratically to the gauge field, one would again have a non-local effective action, indicating that this analytic integration was not a good idea. Perhaps we should step back and explore algorithms before integrating out the fermions.

As Monte Carlo methods are particularly difficult with fermions, perhaps one should look for other numerical methods not based on stochastic processes. Here I include such possibilities as direct diagonalization of the Hamiltonian. So far these methods have been very constrained on possible system size, suggesting that one should look for new approximations to discard irrelevant information as the systems grow in volume.

An extremely difficult unsolved question is the simulation of fermionic systems when the corresponding determinant is not always positive. This situation is of considerable interest because it arises in the study of quark-gluon thermodynamics when a chemical potential is present. All known approaches to this problem are extremely demanding on computer resources. One can move the phase of the determinant into the observables, but then one must divide out the average value of this sign. This is a number which is expected to go to zero exponentially with the lattice volume;, thus, such an algorithm will require computer time growing exponentially with the system size. Another approach is to do an expansion about zero baryon density, but again to get to large chemical potential will require rapidly growing resources. New techniques are badly needed to avoid this growth; hopefully this will be a particularly fertile area for future algorithm development.

REFERENCES:

The group theory here is a variation on what is in M. Creutz, H. Gausterer, and S. Sanielevici, "Biased Monte Carlo algorithms on unitary groups," Phys. Rev. D39, 689-692 (1989).

M. Creutz, "Global Monte Carlo algorithms for many-fermion systems," Phys. Rev. D38, 1228-1238 (1988).