2. MONTE CARLO INTEGRATION

In thermodynamics, as in many other fields of physics, often very high dimensional integrals have to be evaluated. Even in a classical N-body simulation the phase space has dimension $6N$, as there are three coordinates each for the location and position of each particle. In a quantum mechanical problem of N particles the phase space is even exponentially large as a function of N.

2.1. Deterministic integration methods

A Riemannian integral $f(x)$ over an interval $[a,b]$ can be evaluated by replacing it by a finite sum:

$$\int_a^b f(x)dx = \sum_{i=1}^{N} f(a+i\Delta x)\Delta x + \mathrm{O}(\Delta x^2), \tag{1}$$

where $\Delta x = (a-b)/N$. The discretization error decreases as $1/N$ for this simple formula. Better approximations are the trapezoidal rule

$$\int_a^b f(x)dx = \Delta x \left[\frac{1}{2}f(a) + \sum_{i=1}^{N-1} f(a+i\Delta x) + \frac{1}{2}f(b)\right] + \mathrm{O}(\Delta x^2), \tag{2}$$

or the Simpson rule

$$\begin{aligned}\int_a^b f(x)dx &= \frac{\Delta x}{3}\left[f(a) + \sum_{i=1}^{N/2} 4f(a+(2i-1)\Delta x)\right. \\ &\quad \left. + \sum_{i=1}^{N/2-1} 2f(a+2i\Delta x) + f(b)\right] + \mathrm{O}(\Delta x^4),\end{aligned} \tag{3}$$

which scales like N^{-4}.

For more elaborate schemes like the Romberg method or Gaussian integration we refer to textbooks.

In higher dimensions the convergence is much slower though. With N points in d dimensions the linear distance between two points scales only as $N^{-1/d}$. Thus the Simpson rule in d dimensions converges only as $N^{-4/d}$, which is very slow for large d. The solution are Monte Carlo integrators.

2.2. Monte Carlo integrators

With randomly chosen points the convergence does not depend on dimensionality. Using N randomly chosen points $\mathbf{x}_i$ the integral can be approximated by

$$\frac{1}{\Omega}\int f(\mathbf{x})d\mathbf{x} \approx \overline{f} := \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i), \tag{4}$$

where $\Omega := \int d\mathbf{x}$ is the integration volume. As we saw in the previous chapter the errors of such a Monte Carlo estimate the errors scale as $N^{-1/2}$. In $d \geq 9$ dimensions Monte Carlo methods are thus preferable to a Simpson rule.

2.2.1. Importance Sampling

This simple Monte Carlo integration is however not the ideal method. The reason is the variance of the function

$$\mathrm{Var} f = \Omega^{-1} \int f(\mathbf{x})^2 d\mathbf{x} - \left[\Omega^{-1} \int f(\mathbf{x}) d\mathbf{x}\right]^2 \approx \frac{N}{N-1}(\overline{f^2} - \overline{f}^2). \tag{5}$$

The error of the Monte Carlo simulation is

$$\Delta = \sqrt{\frac{\mathrm{Var} f}{N}} \approx \sqrt{\frac{\overline{f^2} - \overline{f}^2}{N-1}}. \tag{6}$$

In phase space integrals the function is often strongly peaked in a small region of phase space and has a large variance. The solution to this problem is "importance sampling", where the points $\mathbf{x}_i$ are chosen not uniformly but according to a probability distribution $p(\mathbf{x})$ with

$$\int p(\mathbf{x}) d\mathbf{x} = 1. \tag{7}$$

Using these p-distributed random points the sampling is done according to

$$\langle f \rangle = \Omega^{-1} \int A(\mathbf{x}) d\mathbf{x} = \Omega^{-1} \int \frac{f(\mathbf{x})}{p(\mathbf{x})} p(\mathbf{x}) d\mathbf{x} \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(\mathbf{x}_i)}{p(\mathbf{x}_i)} \tag{8}$$

and the error is

$$\Delta = \sqrt{\frac{\mathrm{Var} f/p}{N}}. \tag{9}$$

It is ideal to choose the distribution function p as similar to f as possible. Then the ratio f/p is nearly constant and the variance small.

As an example, the function $f(x) = \exp(-x^2)$ is much better integrated using exponentially distributed random numbers with $p(x) = \exp(-\lambda x)$ instead of uniformly distributed random numbers.

A natural choice for the weighting function p is often given in the case of phase space integrals or sums, where an observable O is averaged over all configurations $\mathbf{x}$ in phase space where the probability of a configuration is $p(\mathbf{x})$. The phase space average $\langle O \rangle$ is:

$$\langle O \rangle = \frac{\int O(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}}{\int p(x) d\mathbf{x}}. \tag{10}$$

Here we will want to do importance sampling with the distribution $p(\mathbf{x})$.

2.3. Markov chains and the Metropolis algorithm

In general problems with arbitrary distributions p it will not be possible to create p-distributed configuration from scratch. Instead a Markov process can be used.

Starting from an initial point $\mathbf{x}_0$ a Markov chain of states is generated:

$$\mathbf{x}_0 \to \mathbf{x}_1 \to \mathbf{x}_2 \to \ldots \to \mathbf{x}_n \to \mathbf{x}_{n+1} \to \ldots \tag{11}$$

A transition matrix $W_{\mathbf{xy}}$ gives the transition probabilities of going from state $\mathbf{x}$ to state $\mathbf{y}$ in one step of the Markov process. As the sum of probabilities of going from state $\mathbf{x}$ to any other state is one, the columns of the matrix W are normalized:

$$\sum_{\mathbf{y}} W_{\mathbf{xy}} = 1 \tag{12}$$

A consequence is that the Markov process conserves the total probability. Another consequence is that the largest eigenvalue of the transition matrix W is 1 and the corresponding eigenvector with only positive entries is the equilibrium distribution which is reached after a large number of Markov steps.

We want to determine the transition matrix W so that we asymptotically reach the desired probability $p_{\mathbf{x}}$ for a configuration i. A set of sufficient conditions is:

1. **Ergodicity:** It has to be possible to reach any configuration $\mathbf{x}$ from any other configuration $\mathbf{y}$ in a finite number of Markov steps. This means that for all $\mathbf{x}$ and $\mathbf{y}$ there exists a positive integer $n_0 < \infty$ such that$\forall n > n_0 : (W^n)_{\mathbf{xy}} \neq 0$.
2. **Detailed balance:** The probability distribution $p_{\mathbf{x}}^{(n)}$ changes at each step of the Markov process:
$$\sum_{\mathbf{x}} p_{\mathbf{x}}^{(n)} W_{\mathbf{xy}} = p_{\mathbf{y}}^{(n+1)}. \tag{13}$$
but converges to the equilibrium distribution $p_{\mathbf{x}}$. This equilibrium distribution $p_{\mathbf{x}}$ is an eigenvector with left eigenvalue 1 and the equilibrium condition
$$\sum_{\mathbf{x}} p_{\mathbf{x}} W_{\mathbf{xy}} = p_{\mathbf{y}} \tag{14}$$
must be fulfilled. It is easy to see that the detailed balance condition
$$\frac{W_{\mathbf{xy}}}{W_{\mathbf{yx}}} = \frac{p_{\mathbf{y}}}{p_{\mathbf{x}}} \tag{15}$$
is sufficient.

Having defined W, we now have to construct it explicitly. Let $W^0_{\mathbf{xy}}$ be the probability of proposing a transition from $\mathbf{x}$ to $\mathbf{y}$ and $a_{\mathbf{xy}}$ the probability of accepting it. $1 - a_{\mathbf{xy}}$ corresponds to the probability of rejecting the move. With $W^0_{\mathbf{xy}}$ and $a_{\mathbf{xy}}$ we build $W_{\mathbf{xy}}$ with:

$$W_{\mathbf{xy}} = \begin{cases} W^0_{\mathbf{xy}} a_{\mathbf{xy}} & \text{if } y \neq x \\ W^0_{\mathbf{xx}} + \sum_{z \neq x} W^0_{\mathbf{xz}} (1 - a_{\mathbf{xz}}) & \text{if } y = x \end{cases} \tag{16}$$

Detailed balance can be satisfied by setting

$$a_{\mathbf{xy}} = F\left(\frac{W^0_{\mathbf{yx}}P_y}{W^0_{\mathbf{xy}}P_x}\right). \tag{17}$$

Since

$$a_{\mathbf{yx}} = F\left(\frac{W^0_{\mathbf{xy}}P_x}{W^0_{\mathbf{yx}}P_y}\right) = F\left(\frac{1}{\frac{W^0_{\mathbf{yx}}P_y}{W^0_{\mathbf{xy}}P_x}}\right), \tag{18}$$

the detailed balance condition reduces to:

$$\frac{F(R)}{F(1/R)} = R \text{ where } R = \frac{W^0_{\mathbf{yx}}P_y}{W^0_{\mathbf{xy}}P_x}. \tag{19}$$

There are many possible choices. The Metropolis algorithm [9] is based on the choice:

$$F(R) = \min(R, 1). \tag{20}$$

Thus, one proposes a transition from x to y and accepts it with probability $R = \frac{T^0_{x,y}P_y}{W^0_{\mathbf{xy}}P_x}$.

This is implemented by picking a random number r in the interval $[0:1]$. If $r < R$ ($r > R$) one accepts (rejects) the move. Alternative choices of $F(R)$ are for example:

$$F(R) = \frac{R}{1+R} \tag{21}$$

which is referred to as the heat bath method.

The observable O may now be estimated with:

$$\langle O\rangle_P \approx \frac{1}{N}\sum_{t=1}^{N} O(x_t). \tag{22}$$

2.4. Autocorrelations, equilibration and Monte Carlo error estimates

2.4.1. Autocorrelation effects

In the determination of statistical errors of the Monte Carlo estimates we have to take into account correlations between successive points $\mathbf{x}_i$ in the Markov chain. These correlations between configurations manifest themselves in correlations between the measurements of a quantity O measured in the Monte Carlo process. Denote by O_t the measurement of the observable O evaluated at the t-th Monte Carlo point $\mathbf{x}_t$. The autocorrelations decay exponentially for large time differences Δ:

$$\langle O_t O_{t+\Delta}\rangle - \langle O\rangle^2 \ \mu\ \exp(-\Delta/\tau_O^{(exp)}) \tag{23}$$

Note that the autocorrelation time τ_O depends on the quantity O.

An alternative definition is the integrated autocorrelation time $\tau_O^{(int)}$, defined by

$$\tau_O^{(int)} = \frac{\sum_{\Delta=1}^{\infty} \left(\langle O_t O_{t+\Delta} \rangle - \langle O \rangle^2 \right)}{\langle O^2 \rangle - \langle O \rangle^2} \tag{24}$$

As usual the expectation value of the quantity O can be estimated by the mean:

$$\overline{O} \equiv \frac{1}{N} \sum_i = 1^N O_i \tag{25}$$

The error estimate

$$\Delta O = \sqrt{\frac{\mathrm{VarO}}{N}} \tag{26}$$

has to be modified because consecutive measurements are correlated . The error estimate $(\Delta O)^2$ is calculates as the expectation value of the squared difference between sample average and expectation value:

$$\begin{aligned}
(\Delta O)^2 &= \langle (\overline{O} - \langle O \rangle)^2 \rangle = \langle \left(\frac{1}{N} \sum_{t=1}^{N} O_t - \langle O \rangle \right)^2 \rangle \\
&= \frac{1}{N^2} \sum_{i=1}^{N} \left(\langle O_t^2 - \langle O \rangle^2 \rangle \right) \\
&\quad + \frac{2}{N^2} \sum_{t=1}^{N} \sum_{\Delta=1}^{N-t} \left(\langle O_t O_{t+\Delta} \rangle - \langle O \rangle^2 \rangle \right) \\
&\approx \frac{1}{N} \mathrm{Var} O \, (1 + 2\tau_O^{(int)}) \\
&\approx \frac{1}{N-1} \langle \overline{O^2} - \overline{O}^2 \rangle (1 + 2\tau_O^{(int)})
\end{aligned} \tag{27}$$

In going from the second to third line we assumed $\tau_O^{(int)} \ll N$ and extended the summation over Δ to infinity. In the last line we replaced the variance by an estimate obtained from the sample. We see that the number of statistical uncorrelated samples is reduced from N to $N/(1+2\tau_O^{(int)})$.

In many Monte Carlo simulations the error analysis is unfortunately not done accurately. Thus we wish to discuss this topic here in more detail.

2.4.2. The binning analysis

The binning analysis is a reliable way to estimate the integrated autocorrelation times. Starting from the original series of measurements $O_i^{(0)}$ with $i = 1, \ldots, N$ we iteratively create "binned" series by averaging over to consecutive entries:

$$O_i^{(l)} := \frac{1}{2} \left(O_{2i-1}^{(l-1)} + O_{2i}^{(l-1)} \right), \qquad i = 1, \ldots, N_l \equiv N/2^l. \tag{28}$$

These bin averages $O_i^{(l)}$ are less correlated than the original values $O_i^{(0)}$. The mean value is still the same.

The errors $\Delta O^{(l)}$, estimated incorrectly using equation (26)

$$\Delta O^{(l)} = \sqrt{\frac{\mathrm{Var} O^{(l)}}{N_l - 1}} \approx \frac{1}{N_l} \sqrt{\sum_{i=1}^{N_l} \left(O_i^{(l)} - \overline{O^{(l)}} \right)^2} \tag{29}$$

however increase as a function of bin size 2^l. For $2^l \gg \tau_O^{(int)}$ the bins become uncorrelated and the errors converge to the correct error estimate:

$$\Delta O = \lim_{l \to \infty} \Delta O^{(l)}. \tag{30}$$

This binning analysis gives a reliable recipe for estimating errors and autocorrelation times. One has to calculate the error estimates for different bin sizes l and check if they converge to a limiting value. If convergence is observed the limit ΔO is a reliable error estimate, and $\tau_O^{(int)}$ can be obtained from equation (27) as

$$\tau_O^{(int)} = \frac{1}{2} \left[\left(\frac{\Delta A}{\Delta O^{(0)}} \right)^2 - 1 \right] \tag{31}$$

If however no convergence of the $\Delta O^{(l)}$ is observed we know that $\tau_O^{(int)}$ is longer than the simulation time and we have to perform *much* longer simulations to obtain reliable error estimates.

To be really sure about convergence and autocorrelations it is very important to start simulations always on tiny systems and check convergence carefully before simulating larger systems.

2.4.3. Jackknife analysis

The binning procedure is a straightforward way to determine errors and autocorrelation times for Monte Carlo measurements. For functions of measurements like $U = \langle A \rangle / \langle B \rangle$ it becomes difficult because of error propagation and cross-correlations.

Then the jackknife procedure can be used. We again split the measurements into M bins of size $N/M \gg \tau^{(int)}$ that should be much larger than any of the autocorrelation times.

We could now evaluate the complex quantity U in each of the M bins and obtain an error estimate from the variance of these estimates. As each of the bins contains only a rather small number of measurements N/M the statistics will not be good. The jackknife procedure instead works with $M+1$ evaluations of U. U_0 is the estimate using all bins, and U_i for $i = 1, \ldots M$ is the value when all bins *except* the i-th bin are used. That way we always work with a large data set and obtain good statistics.

The resulting estimate for U will be:

$$U = U_0 - (M-1)(\overline{U} - U_0) \tag{32}$$

with a statistical error

$$\Delta U = \sqrt{M-1}\left(\frac{1}{M}\sum_{i=1}^{M}(U_i)^2 - (\overline{U})^2\right)^{1/2}, \tag{33}$$

where

$$\overline{U} = \frac{1}{M}\sum_{i=1}^{M} U_i, \tag{34}$$

The jackknife analysis is implemented in the ALPS library [10, 11].

2.4.4. Equilibration

Thermalization is as important as autocorrelations. The Markov chain converges only asymptotically to the desired distribution. Consequently, Monte Carlo measurements should be started only after a large number N_{eq} of equilibration steps, when the distribution is sufficiently close to the asymptotic distribution. N_{eq} has to be much larger than the thermalization time which is defined similar to the autocorrelation time as:

$$\tau_O^{(eq)} = \frac{\sum_{\Delta=1}^{\infty}(\langle O_0 O_\Delta\rangle - \langle O\rangle^2)}{\langle O_0\rangle\langle O\rangle - \langle O\rangle^2} \tag{35}$$

It can be shown that the thermalization time is the maximum of all autocorrelation times for all observables and is related to the second largest eigenvalue Λ_2 of the Markov transition matrix W by $\tau^{(th)} = -1/\ln\Lambda_2$. It is recommended to thermalize the system for at least ten times the thermalization time before starting measurements.

3. CLASSICAL MONTE CARLO SIMULATIONS

Before getting to algorithms for quantum systems we first review the corresponding algorithms for classical systems, starting with the Ising model as the simplest model.

3.1. The Ising model

The Ising model is the simplest model for a magnetic system and a prototype statistical system. We will use it for our discussion of thermodynamic phase transitions. It consists of an array of classical spins $\sigma_i = \pm 1$ that can point either up ($\sigma_i = +1$) or down ($\sigma_i = -1$). The Hamiltonian is

$$H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j, \tag{36}$$

where the sum goes over nearest neighbor spin pairs.

Two parallel spins contribute an energy of $-J$ while two antiparallel ones contribute $+J$. In the ferromagnetic case the state of lowest energy is the fully polarized state where all spins are aligned, either pointing up or down.

At finite temperatures the spins start to fluctuate and also states of higher energy contribute to thermal averages. The average magnetization thus decreases from its full value at zero temperature. At a critical temperature T_c there is a second order phase transition to a disordered phase. The Ising model is the simplest magnetic model exhibiting such a phase transition and is often used as a prototype model for magnetism.

The thermal average of a quantity A at a finite temperature T is given by a sum over all states:

$$\langle A \rangle = \frac{1}{Z} \sum_i A_i \exp(-\beta E_i), \tag{37}$$

where $\beta = 1/k_B T$ is the inverse temperature. A_i is the value of the quantity A in the configuration i. E_i is the energy of that configuration.

The partition function

$$Z = \sum_i \exp(-\beta E_i) \tag{38}$$

normalizes the probabilities $p_i = \exp(-\beta E_i)/Z$.

For small systems it is possible to evaluate these sums exactly. As the number of states grows like 2^N a straight-forward summation is possible only for very small N. For large higher dimensional systems Monte Carlo summation/integration is the method of choice.

3.2. The single spin flip Metropolis algorithm

As was discussed in connection with integration it is usually not efficient to estimate the average (37) using simple sampling. The optimal method is importance sampling,

where the states i are not chosen uniformly but with the correct probability p_i, which we can again do using the Metropolis algorithm.

The simplest Monte Carlo algorithm for the Ising model is the single spin flip Metropolis algorithm which defines a Markov chain through phase space.

- Starting with a configuration c_i propose to flip a single spin, leading to a new configuration c'.
- Calculate the energy difference $\Delta E = E[c'] - E[c_i]$ between the configurations c' and c_i.
- If $\Delta E < 0$ the next configuration is $c_{i+1} = c'$
- If $\Delta E > 0$ then $c_{i+1} = c'$ with probability $\exp(-\beta \Delta E)$, otherwise $c_{i+1} = c_i$. We do that by drawing a random number r uniformly distributed in the interval $[0,1[$ and set $c_{i+1} = c'$ if $r < \exp(-\beta \Delta E)$.
- Measure all the quantities of interest in the new configuration.

This algorithm is ergodic since any configuration can be reached from any other in a finite number of spin flips. It also fulfills the detailed balance condition.

3.3. Systematic errors: boundary and finite size effects

In addition to statistical errors due to the Monte Carlo sampling our simulations suffer from systematic errors due to boundary effects and the finite size of the system.

Unless one wants to study finite systems with open boundaries, boundary effects can be avoided completely by using periodic boundary conditions. The lattice is continued periodically, forming a torus. The left neighbor of the leftmost spin is just the rightmost boundary spin, etc..

Although we can avoid boundary effects, finite size effects remain since now all correlations are periodic with the linear system size as period. Here is how we can treat them:

- *Away from phase transitions* the correlation length ξ is finite and finite size effects are negligible if the linear system size $L \gg \xi$. Usually $L > 6\xi$ is sufficient, but this should be checked for each simulation.
- *In the vicinity of continuous phase transitions* we encounter a problem: the correlation length ξ diverges. Finite size scaling can comes to the rescue and can be used to obtain the critical behavior. A detailed discussion of finite size scaling is beyond the scope of these notes.

3.4. Critical behavior of the Ising model

Close to the phase transition at T_c again scaling laws characterize the behavior of all physical quantities. The average magnetization scales as

$$m(T) = \langle |M|/V \rangle \mu \ (T_c - T)^{\beta}, \tag{39}$$

where M is the total magnetization and V the system volume (number of spins).

The magnetic susceptibility $\chi = \frac{dm}{dh}|_{h=0}$ can be calculated from magnetization fluctuations and diverges with the exponent γ:

$$\chi(T) = \frac{\langle M^2/V\rangle - \langle |M|/V\rangle^2}{T} \mu \ |T_c - T|^{-\gamma}. \tag{40}$$

The correlation length ξ is defined by the asymptotically exponential decay of the two-spin correlations:

$$\langle \sigma_0 \sigma_{\mathbf{r}}\rangle - \langle |m|\rangle^2 \mu \ \exp(-r/\xi). \tag{41}$$

It is best calculated from the structure factor $S(\mathbf{q})$, defined as the Fourier transform of the correlation function. For small $\mathbf{q}$ the structure factor has a Lorentzian shape:

$$S(\mathbf{q}) = \frac{1}{1+q^2\xi^2} + \mathrm{O}(q^4). \tag{42}$$

The correlation length diverges as

$$\xi(p) \mu \ |T - T_c|^{-\nu}. \tag{43}$$

At the critical point the correlation function itself follows a power law:

$$\langle \sigma_0 \sigma_{\mathbf{r}}\rangle \mu \ r^{-(d-2+\eta)} \tag{44}$$

where $\eta = 2\beta/\nu - d + 2$.

The specific heat $C(T)$ diverges logarithmically in two dimensions:

$$C(T) \mu \ \ln|T - T_c| \mu \ |T - T_c|^{-\alpha} \tag{45}$$

and the critical exponent $\alpha = 0$.

A good estimate of T_c is obtained from the Binder cumulant

$$U = 1 - \frac{\langle M^4\rangle}{3\langle M^2\rangle^2}, \tag{46}$$

which has a universal value at p_c, also the Binder cumulant has a universal value at T_c. The curves of $U(T)$ for different system sizes L all cross in one point at T_c. This is a consequence of the finite size scaling ansatz:

$$\begin{aligned} \langle M^4\rangle &= (T - T_c)^{4\beta} u_4((T - T_c)L^{1/\nu}) \\ \langle M^2\rangle &= (T - T_c)^{2\beta} u_2((T - T_c)L^{1/\nu}). \end{aligned} \tag{47}$$

Thus

$$U(T,L) = 1 - \frac{u_4((T - T_c)L^{1/\nu})}{3u_2((T - T_c)L^{1/\nu})^2}, \tag{48}$$

which for $T = T_c$ is universal and independent of system size L:

$$U(T_c,L) = 1 - \frac{u_4(0)}{3u_2(0)^2} \tag{49}$$

High precision Monte Carlo simulations actually show that not all lines cross exactly at the same point, but that due to higher order corrections to finite size scaling the crossing point moves slightly, proportional to $L^{-1/\nu}$, allowing a high precision estimate of T_c and ν. For details of the determination of critical points and exponents see e.g. Ref [12, 13].

3.5. "Critical slowing down" and cluster Monte Carlo methods

The importance of autocorrelation becomes clear when we wish to simulate the Ising model at low temperatures. The mean magnetization $\langle m \rangle$ is zero on any finite cluster, as there is a degeneracy between a configuration and its spin reversed counterpart. If, however, we start at low temperatures with a configuration with all spins aligned up it will take extremely long time for all spins to be flipped by the single spin flip algorithm. This problem appears as soon as we get close to the critical temperature, where it was observed that the autocorrelation times diverge as

$$\tau \mu \ [\min(\xi, L)]^z. \tag{50}$$

with a dynamical critical exponents $z \approx 2$ for all local update methods like the single spin flip algorithm.

The reason is that at low temperatures it is very unlikely that even one spin gets flipped, and even more unlikely for a large cluster of spins to be flipped. The solution to this problem in the form of cluster updates was found in 1987 and 1989 by Swendsen and Wang [14] and by Wolff [15]. Instead of flipping single spins they propose to flip big clusters of spins and choose them in a clever way so that the probability of flipping these clusters is large.

3.5.1. Cluster updates

We use the Fortuin-Kastelyn representation of the Ising model, as generalized by Kandel and Domany. The phase space of the Ising model is enlarged by assigning a set $\mathscr{G}$ of possible "graphs" to each configuration C in the set of configurations $\mathscr{C}$. We write the partition function as

$$Z = \sum_{C \in \mathscr{C}} \sum_{G \in \mathscr{G}} W(C, G) \tag{51}$$

where the new weights $W(C, G) > 0$ are chosen such that Z is the partition function of the original model by requiring

$$\sum_{G \in \mathscr{G}} W(C, G) = W(C) := \exp(-\beta E[C]), \tag{52}$$

where $E[C]$ is the energy of the configuration C.

The algorithm now proceeds as follows. First we assign a graph $G \in \mathscr{G}$ to the configuration C, chosen with the correct probability

$$P_C(G) = W(C,G)/W(C). \tag{53}$$

Then we choose a new configuration C' with probability $p[(C,G) \to (C',G)]$, keeping the graph G fixed; next a new graph G' is chosen

$$C \to (C,G) \to (C',G) \to C' \to (C',G') \to \ldots \tag{54}$$

What about detailed balance? The procedure for choosing graphs with probabilities P_G obeys detailed balance trivially. The non-trivial part is the probability of choosing a new configuration C'. There detailed balance requires:

$$W(C,G)p[(C,G) \to (C',G)] = W(C',G)p[(C',G) \to (C,G)], \tag{55}$$

which can be fulfilled using either the heat bath algorithm

$$p[(C,G) \to (C',G)] = \frac{W(C',G)}{W(C,G)+W(C',G)} \tag{56}$$

or by again using the Metropolis algorithm:

$$p[(C,G) \to (C',G)] = \max(W(C',G)/W(C,G),1) \tag{57}$$

The algorithm simplifies a lot if we can find a graph mapping such that the graph weights do not depend on the configuration whenever it is nonzero in that configuration. This means, we want the graph weights to be

$$W(C,G) = \Delta(C,G)V(G), \tag{58}$$

where

$$\Delta(C,G) := \begin{cases} 1 \text{ if } W(C,G) \neq 0, \\ 0 \text{ otherwise.} \end{cases} . \tag{59}$$

Then equation (56) simply becomes $p = 1/2$ and equation (57) reduces to $p = 1$ for any configuration C' with $W(C',G) \neq 0$.

3.5.2. The cluster algorithms for the Ising model

Let us now show how this abstract and general algorithm can be applied to the Ising model. Our graphs will be bond-percolation graphs on the lattice. Spins pointing into the same direction can be connected or disconnected. Spins pointing in opposite directions will always be disconnected. In the Ising model we can write the weights $W(C)$ and $W(C,G)$ as products over all bonds b:

$$W(C) = \prod_b w(C_b) \tag{60}$$

$$W(C,G) = \prod_b w(C_b,G_b) = \prod_b \Delta(C_b,G_b)V(G_b) \tag{61}$$

TABLE 1. Local bond weights for the Kandel-Domany representation of the Ising model.

	$c=\uparrow\uparrow$	$c=\downarrow\uparrow$	$c=\uparrow\downarrow$	$c=\downarrow\downarrow$	V(g)
$\Delta(c,\text{discon.})$	1	1	1	1	$e^{-\beta J}$
$\Delta(c,\text{con.})$	1	0	0	1	$e^{\beta J}-e^{-\beta J}$
w(c)	$\exp(\beta J)$	$\exp(-\beta J)$	$\exp(-\beta J)$	$\exp(\beta J)$	

where the local bond configurations C_b can be one of $\{\uparrow\uparrow,\downarrow\uparrow,\uparrow\downarrow,\downarrow\downarrow\}$

and the local graphs can be "connected" or "disconnected". The graph selection can thus be done locally on each bond.

Table 1 shows the local bond weights $w(c,g)$, $w(c)$, $\Delta(c,g)$ and $V(g)$. It can easily be checked that the sum rule (52) is satisfied.

The probability of a connected bond is $[\exp(\beta J)-\exp(-\beta J)]/\exp(\beta J) = 1-\exp(-2\beta J)$ if two spins are aligned and zero otherwise. These connected bonds group the spins into clusters of aligned spins.

A new configuration C' with the same graph G can differ from C only by flipping clusters of connected spins. Thus the name "cluster algorithms". The clusters can be flipped independently, as the flipping probabilities $p[(C,G)\rightarrow(C',G)]$ are configuration independent constants.

There are two variants of cluster algorithms that can be constructed using the rules derived above.

3.5.3. The Swendsen-Wang algorithm

The Swendsen-Wang or multi-cluster algorithm proceeds as follows:

i) Each bond in the lattice is assigned a label "connected" or "disconnected" according to above rules. Two aligned spins are connected with probability $1-\exp(-2\beta J)$. Two antiparallel spins are never connected.

ii) Next a cluster labeling algorithm, like the Hoshen-Kopelman algorithm is used to identify clusters of connected spins.

iii) Measurements are performed, using improved estimators discussed in the next section.

iv) Each cluster of spins is flipped with probability $1/2$.

3.5.4. The Wolff algorithm

The Swendsen Wang algorithm gets less efficient in dimensions higher than two as the majority of the clusters will be very small ones, and only a few large clusters exist. The Wolff algorithm is similar to the Swendsen-Wang algorithm but builds only one cluster starting from a randomly chosen point. As the probability of this point being on a

cluster of size s is proportional to s the Wolff algorithm builds preferably larger clusters. It works in the following way:

i) Choose a random spin as the initial cluster.
ii) If a neighboring spin is parallel to the initial spin it will be added to the cluster with probability $1-\exp(-2\beta J)$.
iii) Repeat step ii) for all points newly added to the cluster and repeat this procedure until no new points can be added.
iv) Perform measurements using improved estimators.
v) Flip all spins in the cluster.

We will see in the next section that the linear cluster size diverges with the correlation length ξ and that the average number of spins in a cluster is just χT. Thus the algorithm adapts optimally to the physics of the system and the dynamical exponent $z \approx 0$, thus solving the problem of critical slowing down. Close to criticality these algorithms are many orders of magnitudes (a factor L^2) better than the local update methods.

3.6. Improved Estimators

In this section we present a neat trick that can be used in conjunction with cluster algorithms to reduce the variance, and thus the statistical error of Monte Carlo measurements. Not only do these "improved estimators" reduce the variance. They are also much easier to calculate than the usual "simple estimators".

To derive them we consider the Swendsen-Wang algorithm. This algorithm divides the lattice into N_c clusters, where all spins within a cluster are aligned. The next possible configuration is any of the 2^{N_c} configurations that can be reached by flipping any subset of the clusters. The idea behind the "improved estimators" is to measure not only in the new configuration but in all equally probable 2^{N_c} configurations.

As simplest example we consider the average magnetization $\langle m \rangle$. We can measure it as the expectation value $\langle \sigma_{\vec{i}} \rangle$ of a single spin. As the cluster to which the spin belongs can be freely flipped, and the flipped cluster has the same probability as the original one, the improved estimator is

$$\langle m \rangle = \langle \frac{1}{2}(\sigma_{\vec{i}} - \sigma_{\vec{i}}) \rangle = 0. \tag{62}$$

This result is obvious because of symmetry, but we saw that at low temperatures a single spin flip algorithm will fail to give this correct result since it takes an enormous time to flip all spins. Thus it is encouraging that the cluster algorithms automatically give the exact result in this case.

Correlation functions are not much harder to measure:

$$\langle \sigma_{\vec{i}} \sigma_{\vec{j}} \rangle = \begin{cases} 1 & \text{if } \vec{i} \text{ and } \vec{j} \text{ are on the same cluster} \\ 0 & \text{otherwise} \end{cases} \tag{63}$$

To derive this result consider the two cases and write down the improved estimators by considering all possible cluster flips.

Using this simple result for the correlation functions the mean square of the magnetization is

$$\langle m^2 \rangle = \frac{1}{N^2} \sum_{\vec{i},\vec{j}} \langle \sigma_{\vec{i}} \sigma_{\vec{j}} \rangle = \frac{1}{N^2} \langle \sum_c S(c)^2 \rangle, \tag{64}$$

where $S(c)$ is the number of spins in a cluster c. The susceptibility above T_c is simply given by $\beta \langle m^2 \rangle$ and can also easily be calculated by above sum over the squares of the cluster sizes.

In the Wolff algorithm only a single cluster is built. Above sum (64) can be rewritten to be useful also in case of the Wolff algorithm:

$$\begin{aligned} \langle m^2 \rangle &= \frac{1}{N^2} \langle \sum_c S(c)^2 \rangle \\ &= \frac{1}{N^2} \sum_{\vec{i}} \frac{1}{S_{\vec{i}}} S_{\vec{i}}^2 \\ &= \frac{1}{N^2} \sum_{\vec{i}} S_{\vec{i}} = \frac{1}{N} \langle S(\mathrm{c}) \rangle, \end{aligned} \tag{65}$$

where $S_{\vec{i}}$ is the size of the cluster containing the initial site $\vec{i}$. The expectation value for m^2 is thus simply the mean cluster size. In this derivation we replaced the sum over all clusters by a sum over all sites and had to divide the contribution of each cluster by the number of sites in the cluster. Next we can replace the average over all lattice sites by the expectation value for the cluster on a randomly chosen site, which in the Wolff algorithm will be just the one Wolff cluster we build.

Generalizations to other quantities, like the structure factor $S(\vec{q})$ are straightforward. While the calculation of $S(\vec{q})$ by Fourier transform needs at least $O(N \ln N)$ steps, it can be done much faster using improved estimators, here derived for the Wolff algorithm:

$$\begin{aligned} \langle S(\vec{q}) \rangle &= \frac{1}{N^2} \sum_{\vec{r},\vec{r}'} \sigma_{\vec{r}} \sigma_{\vec{r}'} \exp(i\vec{q}(\vec{r}-\vec{r}')) \\ &= \frac{1}{NS(c)} \sum_{\vec{r},\vec{r}' \in cluster} \sigma_{\vec{r}} \sigma_{\vec{r}'} \exp(i\vec{q}(\vec{r}-\vec{r}')) \\ &= \frac{1}{NS(c)} \left| \sum_{\vec{r} \in cluster} \exp(i\vec{q}\vec{r}) \right|^2, \end{aligned} \tag{66}$$

This needs only $O(S(c))$ operations and can be measured directly when constructing the cluster.

Care must be taken for higher order correlation functions. Improved estimators for quantities like m^4 which need at least two clusters and cannot be measured in an improved way using the Wolff algorithm.

3.7. Generalizations of cluster algorithms

Cluster algorithms can be used not only for the Ising model but for a large class of classical, and even quantum spin models. The quantum version is the "loop algorithm", which will be discussed later in the course. In this section we discuss generalizations to other classical spin models.

Before discussing specific models we remark that generalizations to models with different coupling constants on different bonds, or even random couplings are straightforward. All decisions are done locally, individually for each spin or bond, and the couplings can thus be different at each bond.

3.7.1. Potts models

q-state Potts models are the generalization of the Ising model to more than two states. The Hamilton function is

$$H = -J \sum_{\langle i,j \rangle} \delta_{s_i,s_j}, \tag{67}$$

where the states s_i can take any integer value in the range $1,\ldots,q$. The 2-state Potts model is just the Ising model with some trivial rescaling.

The cluster algorithms for the Potts models connect spins with probability $1-e^{-\beta J}$ if the spins have the same value. The clusters are then "flipped" to any arbitrarily chosen value in the range $1,\ldots,q$.

3.7.2. $O(N)$ models

Another, even more important generalization are the $O(N)$ models. Well known examples are the XY-model with $N=2$ and the Heisenberg model with $N=3$. In contrast to the Ising model the spins can point into any arbitrary direction on the N-sphere. The spins in the XY model can point into any direction in the plane and can be characterized by a phase. The spins in the Heisenberg model point into any direction on a sphere.

The Hamilton function is:

$$H = -J \sum_{\langle i,j \rangle} \vec{S}_i \vec{S}_j, \tag{68}$$

where the states $\vec{S}_i$ are $SO(N)$ vectors.

Cluster algorithms are constructed by projecting all spins onto a random direction $\vec{e} \in SO(N)$. The cluster algorithm for the Ising model can then be used for this projection. Two spins $\vec{S}_i$ and $\vec{S}_j$ are connected with probability

$$1 - \exp\left(\min[0, -2\beta J(\vec{e}\cdot\vec{S}_i)(\vec{e}\cdot\vec{S}_j)]\right). \tag{69}$$

The spins are flipped by inverting the projection onto the $\vec{e}$-direction:

$$\vec{S}_i \to \vec{S}_i - 2(\vec{e}\cdot\vec{S}_i)\vec{e}. \tag{70}$$

In the next update step a new direction $\vec{e}$ is chosen.

3.8. The multicanonical ensemble and Wang Landau sampling

While cluster updates can solve critical slowing down at second order phase transitions they are usually inefficient at first order phase transitions and in frustrated systems. Let us consider a first order phase transition, such as in a two-dimensional q-state Potts model with Hamilton function

$$H = -J \sum_{\langle i,j \rangle} \delta_{\sigma_i \sigma_j}, \tag{71}$$

where the spins σ_i can now take the integer values $1, \ldots, q$. For $q > 4$ this model exhibits a first order phase transition, accompanied by exponential slowing down of local single-spin updates. The exponential slow-down is caused by the free energy barrier between the two coexisting meta-stable states at the first order phase transition.

This barrier can be quantified by considering the energy histogram

$$H_{\text{canonical}}[E] = g(E) P_{\text{Boltzmann}}(E) \; \mu \; g(E) \exp(-\beta E) \tag{72}$$

which is the probability of encountering a configuration with energy E during the Monte Carlo simulation. Here

$$g(E) = \sum_c \delta_{E,E(c)} \tag{73}$$

is the density of states. Away from first order phase transitions, $H_{\text{canonical}}[E]$ has approximately Gaussian shape, centered around the mean energy. At first order phase transitions, where the energy jumps discontinuously the histogram $H_{\text{canonical}}[E]$ develops a double-peak structure. The minimum of $H_{\text{canonical}}[E]$ between these two peaks, which the simulation has to cross in order to go from one phase to the other, becomes exponentially small upon increasing the system size. This leads to exponentially large autocorrelation times.

The simplest lattice model showing such a first order thermal phase transition is a two-dimensional Potts model with large q, e.g. $q = 10$. For this model we show in Fig. 1 the probability $P(E,T)$ of visiting a configuration with energy E. This is:

$$P(E,T) = \rho(E) p(E) = \rho(E) e^{-E/k_B T}, \tag{74}$$

where the density of states $\rho(E)$ counts the number of states with energy E. At the critical temperature there are two coexisting phases, the ordered and disordered ones with different energies. In order to tunnel from one to the other one has to continuously change the energy and thus go through the probability minimum between the two peeks. This probability decreases exponentially with system size and we thus have a real problem!

A solution to this tunneling problem are extended ensembles, such as the multicanonical ensemble [16, 17], where the weight of a configuration c is given by

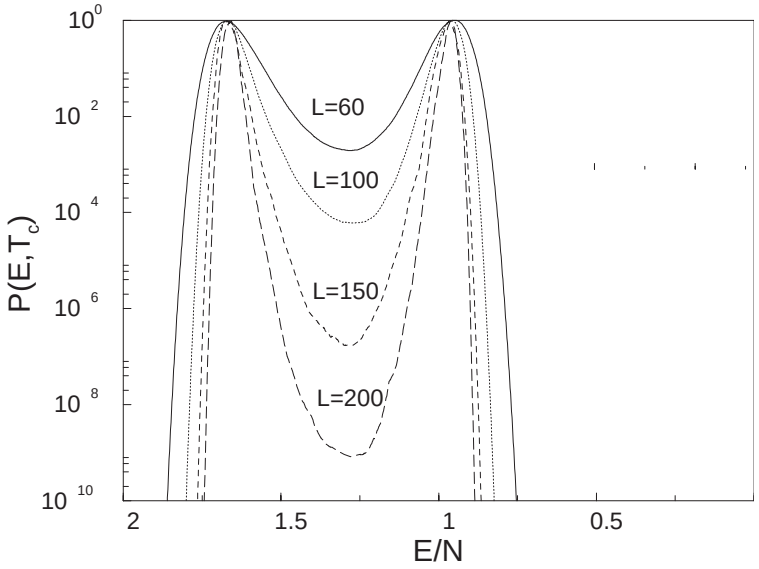

FIGURE 1. The probability $P(E,T) = \rho(E)\exp(-E/k_BT)$ of visiting a state with energy E in a $q = 10$-state Potts model at the critical temperature. The tunneling probability between the two phases (the dip between the two maxima) becomes exponentially small for large systems. This figure is taken from the paper: F. Wang and D.P. Landau, Phys. Rev. Lett. **86**, 2050 (2001).

$P_{\text{multicanonical}}(c)$ μ $1/g(E(c))$ instead of the Boltzmann weight $\exp(-\beta E(c))$. The multicanonical ensemble leads to a flat histogram in energy space

$$H_{\text{multicanonical}}[E] = g(E)P_{\text{multicanonical}}(E) \text{ μ } g(E)\frac{1}{g(E)} = \text{const.} \tag{75}$$

removing the exponentially small minimum. After performing a simulation, measurements in the multicanonical ensemble are reweighted by a factor $P_{\text{Boltzmann}}(E)/P_{\text{multicanonical}}(E)$ to obtain averages in the canonical ensemble.

Since the density of states $g(E)$ and thus the multicanonical weights $P_{\text{multicanonical}}$ are not known initially, a scalable algorithm to estimate these quantities is needed. The Wang-Landau algorithm [18, 19] is a simple iterative method to obtain the density of states $g(E)$ and the multicanonical weights $P_{\text{multicanonical}}(E)$ μ $1/g(E)$. The approach by Wang and Landau is crude but simple. They start with a (very bad guess) $\rho(E) = 1$ for all energies an iteratively improve it:

- Start with $\rho(E) = 1$ and $f = e$
- Repeat
  - Reset a histogram of energies $H(E) = 0$
  - Perform simulations until a histogram of energies $H(E)$ is "flat"
    - pick a random site
    - attempt a local Metropolis update using $p(E) = 1/\rho(E)$
    - increase the histogram at the current energy E: $H(E) \leftarrow H(E) + 1$
    - increase the estimate for $\rho(E)$ at the current energy E: $\rho(E) \leftarrow \rho(E) \cdot f$

  - once $H(E)$ is "flat" (e.g. the minimum is at least 80% of the mean), reduce $f \leftarrow \sqrt{f}$
- stop once $f \approx 1 + 10^{-8}$

As you can see, only a few lines of code need to be changed in your local update algorithm for the Ising model, but a few remarks are necessary:

1. Check for flatness of the histogram not at very step but only after a reasonable number of sweeps N_{sweeps}. One sweep is defined as one attempted update per site.
2. The initial value for f needs to be carefully chosen, $f = e$ is only a rough guide. As discussed in the papers a good choice is picking the initial f such that $f^{N_{\text{sweeps}}}$ is approximately the total number of states (e.g. q^N for a q-state Potts model with N sites).
3. The flatness criterion is quite arbitrary and some research is still necessary to find the optimal choice.
4. The density of states $\rho(E)$ can become very large and easily exceed 10^{10000}. In order to obtain such large numbers the *multiplicative increase* $\rho(E) \leftarrow \rho(E) \cdot f$ is essential. A naive additive guess $\rho(E) \leftarrow \rho(E) + f$ would never be able to reach the large numbers needed.
5. Since $\rho(E)$ is so large, we only store its *logarithm*. The update step is thus $\ln \rho(E) \leftarrow \ln \rho(E) + \ln f$. The Metropolis acceptance probability will be

$$P = \min[1, \exp(\ln \rho(E_{\text{old}}) - \ln \rho(E_{\text{new}}))] \tag{76}$$

Another advantage of the Wang-Landau algorithm is that, once we know the density of states $\rho(E)$, we can directly calculate the partition function

$$Z = \sum_c E_c e^{-E_c/k_B T} = \sum_E \rho(E) e^{-E/k_B T} \tag{77}$$

and the free energy

$$F = -k_B T \ln Z = -k_B T \ln \sum_E \rho(E) e^{-E/k_B T} \tag{78}$$

which are both not directly accessible in any other Monte Carlo algorithm. All other thermodynamic properties such as the susceptibility or the specific heat can now be calculated simply as derivatives of the free energy.

Recent investigations have shown that a flat histogram, as obtained by the multicanonical ensemble, is not optimal but still shows signs of critical slowing down [20]. Optimized ensembles, assigning larger weights to configurations in the critical region, can be found and lead to further improvements in the efficiency of algorithms by several orders of magnitude [21].

exp()

REFERENCES

10. F. Alet, P. Dayal, A. Grzesik, M. Honecker, A. Laeuchli, S. R. Manmana, I. P. McCulloch, F. Michel, R. M. Noack, G. Schmid, U. Schollwoeck, S. Stoeckli, S. Todo, S. Trebst, M. Troyer, P. Werner, and S. Wessel, *J. Phys. Soc. Jap. Suppl.* **74**, 30 (2005).
11. A. Albuquerque, F. Alet, P. Corboz, P. Dayal, A. Feiguin, S. Fuchs, L. Gamper, E. Gull, S. Gürtler, A. Honecker, R. Igarashi, M. Körner, M. Kozhevnikov, A. Läuchli, S. Manmana, M. Matsumoto, I. McCulloch, F. Michel, R. Noack, G. Pawlowski, L. Pollet, T. Pruschke, U. Schollwöck, S. Todo, S. Trebst, M. Troyer, P. Werner, and S. Wessel, *J. Mag. Mag. Mat.* **310**, 1187 (2007).
12. A. M. Ferrenberg, and D. P. Landau, *Phys. Rev. B.* **44**, 5081 (1991).
13. K. Chen, A. M. Ferrenberg, and D. P. Landau, *Phys. Rev. B.* **48**, 3249 (1993).
14. R. Swendsen, and J.-S. Wang, *Phys. Rev. Lett.* **58**, 86–89 (1987).
15. U. Wolff, *Phys. Rev. Lett.* **62**, 361–364 (1989).
16. B. A. Berg, and T. Neuhaus, *Phys. Lett. B* **267**, 249–253 (1991).
17. B. A. Berg, and T. Neuhaus, *Phys. Rev. Lett.* **68**, 9–12 (1992).
18. F. Wang, and D. Landau, *Phys. Rev. Lett.* **86**, 2050–2053 (2001).
19. F. Wang, and D. Landau, *Phys. Rev. E* **64**, 056101 (2001).
20. P. Dayal, S. Trebst, S. Wessel, D. Würtz, M. Troyer, S. Sabhapandit, and S. N. Coppersmith, *Phys. Rev. Lett.* **92**, 097201 (2004).
21. S. Trebst, D. Huse, and M. Troyer, *Phys. Rev. E* **70**, 046701 (2004).
22. N. V. Prokof'ev, B. V. Svistunov, and I. S. Tupitsyn, *JETP* **87**, 310–321 (1998).
23. A. Sandvik, and J. Kurkijärvi, *Phys. Rev. B* **43**, 5950–5961 (1991).
24. D. Handscomb, *Proc. Cambridge Philos. Soc.* **58**, 594–598 (1962).
25. A. W. Sandvik, *Phys. Rev. B* **56**, 11678–11690 (1997).
26. E. L. Pollock, and D. M. Ceperley, *Phys. Rev. B* **36**, 8343–8352 (1987).
27. M. Jarrell, and J. Gubernatis, *Physics Reports* **269**, 133–195 (1996).