Physics 514 – Magnetic Systems

Emanuel Gull

October 29, 2015

Chapter 1

Magnetism

1.1 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 behavior of the system is governed by a Hamiltonian

$$H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j, \tag{1.1}$$

where the brackets $\langle \cdot \rangle$ denote a summation over nearest neighbor spin pairs.

Two parallel spins contribute an energy of $-J$ while two antiparallel ones contribute $+J$. In the ferromagnetic case there are two states of lowest energy: the fully polarized states where all spins either point 'up' or all spins point 'down'.

At finite temperature 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, similar to the percolation transition we discussed in the percolation problem.

The Ising model is the simplest model exhibiting such a phase transition and is often used as a prototype model for magnetism. To discuss the phase transition the scaling hypothesis introduced in the context of percolation will again be used.

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

$$\langle A \rangle = \frac{1}{Z} \sum_i A_i \exp(-\beta E_i) \tag{1.2}$$

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 or Zustandssumme

$$Z = \sum_i \exp(-\beta E_i) \tag{1.3}$$

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

For small systems it is possible to evaluate these sums exactly by enumerating all possible configurations. 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.

In two dimensions (and at zero field), the problem can be solved analytically. The calculation, first done by Onsager in 1944, is rather involved, but is sometimes taught in advanced statistical mechanics courses. The (exact) critical exponents of this solution are $\beta = 1/8$, $\gamma = 7/4$, $\nu = 1$, and $\alpha = 0$.

1.2 The single spin flip Metropolis algorithm

Direct Monte Carlo sampling of configurations using random configurations is not efficient: a random configuration has typically very high energy and is far away from the configurations which are relevant at typical temperatures (e.g. temperatures around T_c.) Importance sampling, where configurations are generated with the correct probability $p_i = \exp(-\beta E_i)$ instead of a uniform probability, is much more efficient. We cannot directly generate configurations with that probability, but we can use the Metropolis algorithm to do so using a Markov chain.

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

1. Start with a configuration c_i, propose to flip a single spin, leading to a new configuration c'.

2. Calculate the energy difference $\Delta E = E[c'] - E[c_i]$ between the configurations c' and c_i.

3. If $\Delta E < 0$ the next configuration is $c_{i+1} = c'$.

4. 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 $r < \exp(-\beta \Delta E)$.

5. 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.

You will implement this algorithm in preparation for the projects.

1.3 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. In contrast to the percolation problems, boundary effects can be avoided completely by using periodic boundary conditions. The lattice is the 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 period of the linear system size. In the context of the percolation problem we have already learned how to deal with finite size effects:

- *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 the same problem as in percolation: the correlation length ξ diverges. Again, finite size scaling comes to the rescue and we can obtain the critical behavior as discussed in the chapter on percolation.

1.4 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

$$m(T) = \langle |M|/V \rangle \propto (T_c - T)^{\beta}, \tag{1.4}$$

where M is the total magnetization and V the system volume (number of spins). Note that on any finite system $\langle M \rangle = 0$ due to symmetry: for any configuration with non-zero magnetization M there exists another configuration which has the same energy but all spins flipped and therefore $-M$.

The magnetic susceptibility, the quantity characterizing magnetic fluctuations,

$$\chi = \left. \frac{dm}{dh} \right|_{h=0} \tag{1.5}$$

Response to an external field (h)

can be calculated from the magnetization fluctuations and diverges with the exponent γ:

$$\chi(T) = \frac{\langle M^2/V \rangle - \langle |M| \rangle^2 / V}{T} \propto |T_c - T|^{-\gamma} \tag{1.6}$$

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

$$\langle \sigma_0 \sigma_r \rangle - \langle |m| \rangle^2 \propto \exp(-r/\xi). \tag{1.7}$$

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

$$S(q) = \frac{1}{1 + q^2 \xi^2} + O(q^4) \tag{1.8}$$

The correlation length diverges as

$$\xi(T) \propto |T - T_c|^{-\nu} \tag{1.9}$$

At the critical point the correlation function again follows the same power law as in the percolation problem:

$$\langle \sigma_0 \sigma_r \rangle \propto r^{-(d-2+\eta)}, \tag{1.10}$$

where $\eta = 2\beta/\nu - d + 2$, derived from the same scaling laws as in percolation.

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

$$C(T) \propto \log |T - T_c| \propto |T - T_c|^{-\alpha}, \tag{1.11}$$

with critical exponent $\alpha = 0$.

Like in percolation, finite size scaling is the method of choice for the determination of these exponents.

A good estimate of T_c is obtained from the Binder cumulant

$$U = 1 - \frac{\langle M^4 \rangle}{3 \langle M^2 \rangle^2}. \tag{1.12}$$

Just as the function $\Pi(L, p)$ in the percolation problem 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:

$$\langle M^4 \rangle = (T - T_c)^{4\beta} u_4((T - T_c) L^{1/\nu}) \tag{1.13}$$

$$\langle M^2 \rangle = (T - T_c)^{2\beta} u_2((T - T_c) L^{1/\nu}), \tag{1.14}$$

and as a consequence thereof

All system size curves will behave the same way at the critical point!

$$U(T, L) = 1 - \frac{u_4((T - T_c) L^{1/\nu})}{3 u_c((T - T_c) L^{1/\nu})^2}, \tag{1.15}$$

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

$$U(T_c, L) = 1 - \frac{u_4(0)}{3 u_2(0)^2}. \tag{1.16}$$

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. More information can be found in the literature[1]

1.5 Critical slowing down and cluster Monte Carlo methods

The importance of autocorrelation effect becomes clear when we wish to simulate the Ising model at low temperature. The mean magnetization $\langle m \rangle$ is zero on any finite cluster. However, if we start at low temperature with a configuration which has all spins pointing 'up' it will take an 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

$$\tau \propto [\min(\xi, L)]^z \tag{1.17}$$

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

The reason is that at low temperature 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 was found in 1987 and 1989 by Swendsen and Wang[2] and by Wolff[3]: 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.

1.5.1 Kandel-Domany framework

We use the Fortuin - Kastel representation of the Ising model, as generalized by Kandel and Domany. The phase space of the Ising model is enlarged by introducing additional degrees of freedom: we assign a set $\mathcal{G}$ of graphs to each configuration C in the set of configurations $\mathcal{C}$. We write the partition function as

$$Z = \sum_{C \in \mathcal{C}} \sum_{G \in \mathcal{G}} W(C, G), \tag{1.18}$$

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

$$\sum_{G \in \mathcal{G}} W(C, G) = W(C) := \exp(-\beta E[C]) \tag{1.19}$$

[1] See, e.g., A. M. Ferrenberg and D. P. Landau, Phys. Rev. B **44** 5081 (1991); K. Chen, A. M. Ferrenberg and D. P. Landau, Phys. Rev. B **48** 3249 (1993) and recent editions of the book by Landau and Binder

[2] R.h. Swendsen and J.S. Wang, Phys. Rev. Lett. **58**, 86 (1987)

[3] U. Wolff, Phys. Rev. Lett. **62**, 361

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

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

$$P_C(G) = W(C,G)/W(C) \tag{1.20}$$

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, and so on:

$$C \to (C,G) \to (C',G) \to C' \to (C',G') \to \dots \tag{1.21}$$

The procedure for choosing graphs with probabilities $P_C(G)$ obeys detailed balance trivially. The non-trivial part is the probability of choosing a new configuration C'. The detailed balance condition is

$$W(C,G)p[(C,G) \to (C',G)] = W(C',G)p[(C',G) \to (C,G)], \tag{1.22}$$

which we can fulfill using either the heat bath algorithm

$$p[(C,G) \to (C',G)] = \frac{W(C',G)}{W(C,G) + W(C',G)} \tag{1.23}$$

or, again, by using the Metropolis algorithm:

$$p[(C,G) \to (C',G)] = \min(W(C',G)/W(C,G), 1) \tag{1.24}$$

The algorithm simplifies a lot if we can find a graph mapping such that the graph weights to not depend on the configuration, whenever they are non-zero in that configuration. This means we want the graph weights to be

$$W(C,G) = \Delta(C,G)V(G) \tag{1.25}$$

where V is only dependent on the graph and

$$\Delta(C,G) := \begin{cases} 1 & \text{if } W(C,G) \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{1.26}$$

Then equation 1.23 simply becomes $p = 1/2$ and eq. 1.24 reduces to $p = 1$ for any configuration C' with $W(C',G) \neq 0$.

1.5.2 Cluster algorithms for the Ising model

This abstract and general algorithm can be applied to the Ising model. Our graphs G will be bond-percolation graphs connected 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:

$$W(C) = \prod_b w(C_b) \tag{1.27}$$

$$W(C,G) = \prod_b w(C_b, G_b) = \prod_b \Delta(C_b, G_b)V(G_b) \tag{1.28}$$

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

Table 1.1: Choice of local bond weights in the Kandel Domany representation of the Ising model.

where the local bond configurations C_b can be one of the four possible combinations $\uparrow\uparrow, \uparrow\downarrow, \downarrow\uparrow, \downarrow\downarrow$, and the local graphs can be either 'connected' or 'disconnected'. The graph selection can thus be done locally on each bond. Table 1.1 shows the local bond weights $w(c, g), w(c), \Delta(c, g)$ and $V(g)$. The sum rule 1.19 is satisfied. The probability of a connected bond is

$$p_c = \begin{cases} \frac{\exp(\beta J) - \exp(-\beta J)}{\exp(\beta J)} = 1 - \exp(2\beta J), & \text{if two spins aligned} \\ 0 & \text{otherwise.} \end{cases} \tag{1.29}$$

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 $p[(C, G) \to (C', G)]$ are configuration independent constants.

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

1.5.3 The Swendsen-Wang algorithm

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

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

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

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

4. Each cluster of spins is flipped with probability 1/2.

1.5.4 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 preferentially builds larger clusters. It works in the following way:

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

The linear cluster size diverges with the correlation length ξ and 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 is $z \approx 0$, thus solving the problem of critical slowing down. Close to criticality these algorithms are many orders of magnitude (a factor L^2) better than the local update methods. Away from criticality sometimes a hybrid method mixing cluster updates and local updates can be the ideal method.

1.6 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 first 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.

The simplest example is the average magnetization $\langle m \rangle$. We can measure it as the expectation value $\langle \sigma_i \rangle$ 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

$$\langle m \rangle = \left\langle \frac{\sigma_i - \sigma_{i'}}{2} \right\rangle = 0. \tag{1.30}$$

This result is obvious because of symmetry, but we saw that at low temperature a single spin flip algorithm will fail to give the correct result since it takes an exponentially long time to flip all spins.

Correlation functions are not much harder to measure:

$$\langle \sigma_i \sigma_j \rangle = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are on the same cluster} \\ 0 & \text{otherwise.} \end{cases} \tag{1.31}$$

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

$$\langle m^2 \rangle = \frac{1}{N^2} \sum_{ij} \langle \sigma_i \sigma_j \rangle = \frac{1}{N^2} \langle \sum_{cluster} S(cluster)^2 \rangle, \tag{1.32}$$

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

In the case of the Wolff algorithm only a single cluster is built. The expectation value of the square of the magnetization is (replacing the sum over all clusters by a sum over all sites, and then replacing the average over all lattice sites by the expectation value for the cluster on a randomly chosen site)

$$\langle m^2 \rangle = \frac{1}{N^2} \langle \sum_{cluster} S(cluster)^2 \rangle \tag{1.33}$$

$$= \frac{1}{N^2} \sum_i \frac{1}{S(\text{cluster containing } i)} S(\text{cluster containing } i)^2 \tag{1.34}$$

$$= \frac{1}{N^2} \sum_i S(\text{cluster containing } i) = \frac{1}{N^2} \langle S(\text{cluster}) \rangle. \tag{1.35}$$

The expectation value for m^2 is thus simply the mean cluster size.

Generalizations to other quantities are straightforward. While the calculation of the structure factor $S(q)$ needs at least $O(NlogN)$ steps using FFTs, it can be done much faster using improved estimators. For the Wolff algorithm:

$$\langle S(q) \rangle = \frac{1}{N^2} \sum_{rr'} \sigma_r \sigma_{r'} \exp(iq(r - r')) \tag{1.36}$$

$$= \frac{1}{NS(\text{cluster})} \sum_{rr' \in cluster} \sigma_r \sigma_{r'} \exp(iq(r - r')) \tag{1.37}$$

$$= \frac{1}{NS(\text{cluster})} \left| \sum_{r \in cluster} \exp(iqr) \right|^2. \tag{1.38}$$

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

Higher order correlation functions, *e.g.* quantities like m^4, contain terms of the form $\langle S(cluster_1) S(cluster_2) \rangle$. These cannot be measured in algorithms that only generate one cluster, *i.e.* they have to be measured using conventional estimators.

1.7 Generalizations of cluster algorithms

There are a large class of classical and quantum spin models for which cluster updates exist. Here we discuss generalizations to classical spin models.

A common class of models are models that have different coupling constants for different bonds, or even random couplings. The generalization of these cluster methods to such systems is straightforward. All decisions are done locally, individually for each spin or bond, and the couplings can thus be different at each bond.

1.7.1 Potts model

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

$$H = -J \sum_{\langle i,j \rangle} \delta_{s_i s_j} \tag{1.39}$$

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

The cluster algorithms for the Potts model connect spins with probability $1 - \exp(-\beta J)$ if the spins have the same value. The clusters are then 'flipped' to an arbitrarily chosen value in the range $1, \ldots, q$.

1.7.2 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

$$H = -J \sum_{\langle i,j \rangle} S_i S_j, \tag{1.40}$$

where the states S_i are N-dimensional unit vectors.

Cluster algorithms are constructed by projecting all the spins onto a random direction $\hat{e}$. The cluster algorithm for the Ising model can then be used for this projection. Two spins S_i and S_j are connected with probability

$$1 - \exp(\min[0, -2\beta J(\hat{e}\dot{S}_i)(\hat{e}\dot{S}_j)]). \tag{1.41}$$

The spins are flipped by inverting the projection on the $\hat{e}$ direction:

$$S_i \to S_i - 2(\hat{e}\dot{S}_i)\hat{e} \tag{1.42}$$

In the next update step a new random direction $\hat{e}$ on the unit sphere is chosen.

The Heisenberg model

In two dimensions, models with a continuous symmetry, like the $O(N)$ models with $N \geq 2$ do not exhibit a phase transition at a finite critical temperature, as is proven by the Mermin-Wagner theorem. The reason are the Goldstone modes, long wavelength spin waves that have vanishingly low excitation energies. As a consequence the two-dimensional Heisenberg model has $T_c = 0$ and exponentially growing correlation length at low temperatures $\xi \propto \exp(2\pi J/T)$.

The 2d XY model

The only exception to the rule that models with $N \geq 2$ do not exhibit any finite temperature phase transition in two dimensions is the XY model, which has a finite-T Kosterlitz Thouless transition.

This is a phase transition where there is no non-zero magnetization at any non-zero temperature. However, the vorticity remains finite up to a critical temperature $T_c > 0$. At T_c it jumps from the universal value $2T_c/\pi$ to zero.

1.8 The Wang-Landau algorithm

While the cluster algorithms discussed in this section solve the problem of critical slowing down at second order (continuous) phase transitions, they do not help at all at first order phase transitions. At first order phase transitions there is no divergence of any quantity and both the disordered and the ordered phase remain (meta-) stable throughout the transition. The most famous example is the solid/liquid transition, where both water and ice coexist. Water remains liquid even when cooled below the freezing temperature, until some ice crystal nucleates in the super-cooled water. There is a large free energy, the surface energy of the first ice crystal, which has to be overcome before freezing sets in. This leads to macroscopically large tunneling times between the two coexisting phases at the phase transition.

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

$$P(E,T) = \rho(E)p(E) = \rho(E)e^{-E/k_BT}, \tag{1.43}$$

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 one and the disordered one, 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 peaks. This probability decreases exponentially with system size!

The Wang-Landau algorithm has been very successful at solving this problem. It starts from the simple observation that the probability minimum vanishes if we choose $p(E) \propto 1/\rho(E)$ instead of the Boltzmann weight $p(E) \propto$

$\exp(-E/k_BT)$:

$$P(E,T) = \rho(E)p(E) \propto \rho(E)/\rho(E) = const. \tag{1.44}$$

During our sampling we thus get a 'flat histogram' in energy.

Generalizations of this algorithm construct flat histograms in other quantities (usually called 'reaction coordinates'): distance, series expansion order, etc... even two-dimensional flat histograms can be constructed. The algorithm has enjoyed much popularity over the last years in the simulation of quantum and classical (in particular biological) systems with barriers in phase space.

1.8.1 Determining $\rho(E)$

The only problem is that $\rho(E)$ is not known, since it requires the solution of the problem. The approach by Wang and Landau generates it dynamically in the course of the simulation:

1. Start with $\rho(E) = 1$ and $f = e$
2. Collect a histogram of energies $H(E)$ and set it, initially, to zero.
3. Perform simulations until a histogram of energies $H(E)$ is 'flat'.
   (a) pick a random site
   (b) attempt a Metropolis update using $p(E) = 1/\rho(E)$ and the single spin flip algorithm
   (c) increase the histogram at the current energy E: $H(E) \leftarrow H(E) + 1$
   (d) increase the estimate for $\rho(E)$ at the current energy E: $\rho(E) \leftarrow \rho(E)f$
4. once $H(E)$ is flat, e.g. the minimum is at least 80% of the mean, reduce $f \leftarrow \sqrt{f}$.
5. Set the histogram back to zero and repeat. Stop once $f \approx 1 + 10^{-8}$.

Only a few lines of code need to be changed in the local update algorithm for the Ising model. A few remarks:

1. The flatness criterion of 80% is arbitrary. Different choices of it will yield fewer or more steps until f is reduced, but the end result will be independent of it. 80% is a frequently used default value.
2. Check for flatness of H after a reasonable number of Monte Carlo 'sweeps', where one sweep is defined as one update per site.
3. The initial value for f needs to be carefully chosen. $f = e$ is a rough guide, check the literature!

4. The density of states $\rho(E)$ can become very large and easily exceed 2^{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 such large numbers.

5. Since $\rho(E)$ is so large, we only store $\log \rho(E)$. The update step is $\log \rho(E) \leftarrow \log \rho(E) + \log f$. The Metropolis acceptance probability is:

$$P = \min[1, \exp(\log \rho(E_{\text{old}}) - \log \rho(E_{\text{new}}))] \tag{1.45}$$

1.8.2 Calculating Thermodynamic Properties

Another advantage of the Wang Landau algorithm is that a range of observable can be computed directly, once the density of states $\rho(E)$ is known. For example we have direct access to the partition function

$$Z = \sum_c \exp\left(\frac{-E(c)}{k_B T}\right) = \sum_E \rho(E) e^{-E/k_B T} \tag{1.46}$$

and, knowing the partition function, to the free energy

$$F = -k_B T \log Z = -k_B T \log \sum_E \rho(E) \exp\left(\frac{-E}{k_B T}\right) \tag{1.47}$$

Both of these quantities are usually not directly accessible in Monte Carlo. Susceptibilities, specific heat, and other thermodynamic quantities can now be written as derivatives of the free energy.