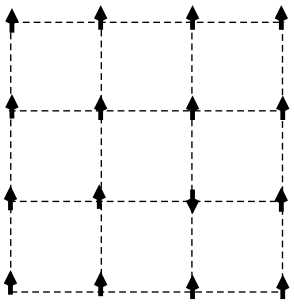
From "Simple" to "Art": more elaborate and efficient methods.

There are cases when the single spin-flip Metropolis scheme is so slow and inefficient (in terms of its autocorrelation time) that it fails to produce answers with the required accuracy. Below we look at some of the methods which outperform the spin-flip algorithm by orders of magnitude in certain regions of parameter space, especially in the vicinity of continuous phase transition points.

What the right data structure can do.

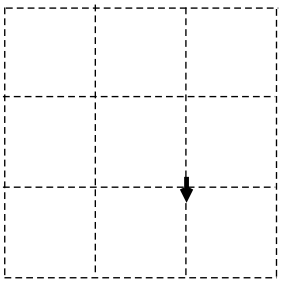
Single spin-flip algorithm works fine at high temperature where there no correlations between spins.

At very low temperature when most spins point in the same direction it makes little sense to select spins at random and suggest to flip them because the typical acceptance ratio will be exponentially small, $R \sim \exp\{-2d \cdot 2J/T\} \ll 1$.



The low-T phase of the Ising model presents a nice example illustrating importance of the proper data structure. If we describe the spin configuration as a simple array $\{\sigma_i\}$, $i = 1, 2, \ldots, N$, we have to store N numbers and update them. But when most of them are $\sigma_i = 1$, the array is "boring". Instead, we may store in memory an array $\{i_l\}$, $l = 1, 2, \ldots, M$, where i_l are sites with $\sigma_{i_l} = -1$

and understand (without mentioning this explicitly) that all other spins are up. The same configuration as above is now described by the following picture. Given new data structure describing low-T configurations we have to update only M and $\{i_l\}$ numbers $\rightarrow$ an exponentially smaller amount of information.



The efficient algorithm follows immediately then, and this is how one can do it. We will need two updates U_+ and U_-, one increasing the number of

down-spins, $M \to M+1$, and the other decreasing it, $M \to M-1$. At each MC step we decide whether to use update U_+ or U_- with probabilities p_+ and $p_- = 1 - p_+$ correspondingly.

- In U_+, select at random any site which is not in the $\{i_l\}$-list, and suggest to flip it. A simple way to implement this efficiently is to introduce an auxiliary array of logical variables, $up(N)$, such that $up(i)$ =.true. if $\sigma_i = 1$ and $up(i)$ =.false. otherwise. To select the up-spin we generate $I = [rndm() * N] + 1$, and check the value of $up(I)$ until it is .true. If the update is accepted, we add I to the $\{i_l\}$-list as the $M+1$-st entry and increase $M = M + 1$.

- In U_-, we select at random any spin from the $\{i_l\}$-list and suggest to flip it. If this update is accepted, we drop the corresponding site from the $\{i_l\}$-list and decrease $M = M - 1$.

Tiny-tiny-tiny Appendix : Updating lists is easy:
- adding entry I: $\quad M = M + 1; \quad i_M = I$
- removing entry i_I: $\quad i_I = i_M; \quad M = M - 1$

To complete the scheme we need the formula for the acceptance ratio. The balance Eq. is

$$p_+ \, \frac{1}{N-M} \, e^{-\beta E_M} \, P_{acc}^{(+)} = p_- \, \frac{1}{M+1} \, e^{-\beta E_{M+1}} \, P_{acc}^{(-)} \, , \tag{1}$$

where term-by-term we write the probabilities of using a particular update, selecting a given site, configuration weight, and accepting the update. Thus

$$\frac{P_{acc}^{(+)}}{P_{acc}^{(-)}} = R = \frac{p_-}{p_+} \, \frac{N-M}{M+1} \, e^{-\beta(E_{M+1} - E_M)} \, . \tag{2}$$

Important reminder! In Eq. (2) the value of M is the number of down-spins before the U_+ update is implemented. Since the value of M is updated by the algorithm, the same expression will look like

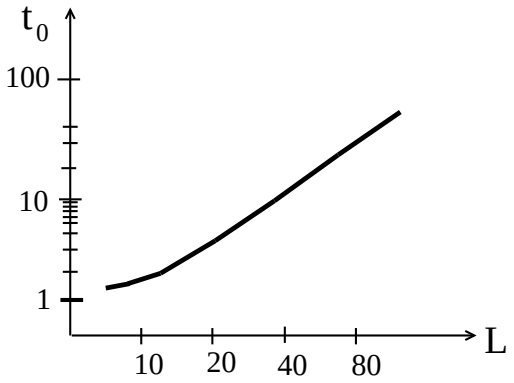
$$\frac{P_{acc}^{(-)}}{P_{acc}^{(+)}} = \frac{1}{R} = \frac{p_+}{p_-} \, \frac{M}{N-(M-1)} \, e^{\beta(E_M - E_{M-1})} \, , \tag{3}$$

when the acceptance ratio is calculated for going backwards using the U_- update with M understood as the current configuration parameter.

Note that at low temperature $N \gg M \approx Ne^{-4dJ/T}$, which means that for $p_+ = p_- = 1/2$ the acceptance ratio will be very close to unity most of the time (except rare cases when we update n.n. of down-spins). The efficiency of the low-T spin-flip algorithm is in addressing up and down spins separately, in fact, we suggest to flip down spins far more frequently then up-spins as $T \to 0$. The lesson to learn is that efficient algorithms should address the "important" degrees of freedom frequently. In the low-T phase of the Ising model such degrees of freedom are isolated down-spins. Our "advanced" spin-flip scheme works reasonably well at high temperature too when $M \approx N/2$, but it has no efficiency gain over the more simple scheme discussed previously.

The real disaster happens when we start simulating critical properties of the model, i.e. in the immediate vicinity of the critical point T_c, or at $|T - T_c|/T_c \ll 1$. In the critical region the system is developing long-range correlations between the spins and the important degrees of freedom are fluctuating fractal domains of similarly oriented spins. This picture has very little to do with the individual values of spins and, correspondingly, single spin-flip algorithms are <u>not</u> updating the relevant (collective in nature) degrees of freedom efficiently. To quantify how algorithm works we look at the autocorrelation time behavior, e.g. for energy, magnetization modulus, or magnetic susceptibility at T_c for different system sizes.

Even when t_0 is measured in sweeps, the spin-flip algorithm gives $t_0(T_c, L) \sim L^z$, where z is called the dynamical critical exponent of the algorithm.



The divergence of t_0 with the system size means that it takes longer and longer (even disregarding the scaling of the time unit = sweep = L^d updates) for the algorithm to produce an uncorrelated configuration. The dynamical critical exponent for the spin-flip algorithm is about $z \approx 2$ in two dimensions. It is possible to simulate square systems with $L = 10^4$; the large value of

z tells us then that the efficiency will drop by a factor of 10^{6-7} for the largest system size. Obviously, we face a serious problem here (it is called the **"critical slowing down"** problem) and should seek a better algorithm for updating configurations close to T_c.

The idea of having a better data structure also helps to simulate the COP Ising model at low temperature, or for M_0 close to saturation. Instead of selecting a pair of opposite-direction spins at random and rejecting most selections (because at low T randomly selected pairs will be same-direction spins with probability close to unity) we may select σ_1 from the list of down-spins and σ_2 from the list of up-spins. In this case the number of up- and down-spins does not change in the update (we still have to update the corresponding lists by exchanging their entries) and the probability of selecting a given pair is exactly same in ν and ν'. Although the acceptance ratio from the previous Section does not change, $R = e^{-\beta(E_{\nu'} - E_\nu)}$, no time is wasted on the selection process—all pairs qualify for the update.

Classical cluster algorithms: Swendsen-Wang, Wolff and Niedermayer, approaches

The original idea of cluster algorithms was introduced by Swendsen and Wang and later slightly modified by Niedermayer and Wolff. I will discuss the Wolff's implementation first. Well, if relevant degrees of freedom close to T_c are domains of similarly oriented spins then we should update domains as a whole, not single spins. Easy to say then to do it properly without violating the balance Eq.! However, the solution proposed by Swendsen and Wang is simple and thus brilliant (Wolff's version):

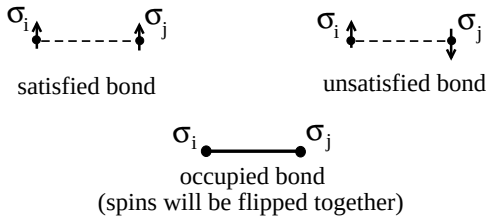
1 Select any spin in the lattice at random; let it be σ_I. This is the first spin of our future domain, or cluster, which I will denote by $\mathcal{C}$ for brevity.

2 Look at all n.n. of spin σ_I, and if some of them point in the same direction then include them into $\mathcal{C}$ with probability P_{add}. Make a list of all <u>new</u> members of $\mathcal{C}$.

3 For each new spin added to $\mathcal{C}$ examine its n.n. which are not part of $\mathcal{C}$, and if they point in the direction of σ_I then include them into the cluster with probability P_{add}. Make a list of new members of $\mathcal{C}$...

4 The construction stops when the new member list is empty :(
Flip all spins in $\mathcal{C}$ with probability R.

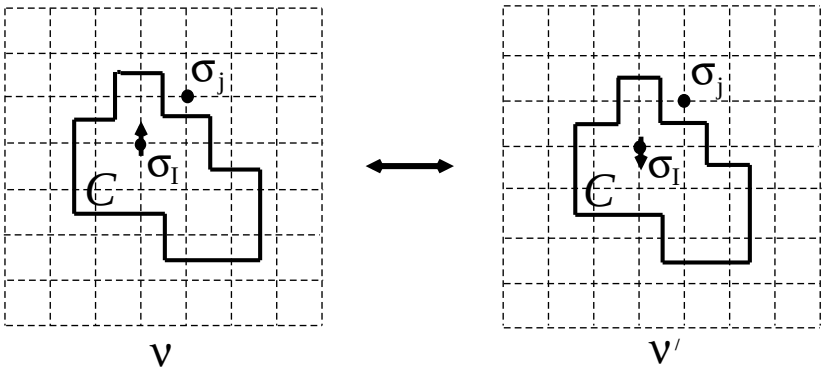
There is a convenient language of bond states to describe cluster algorithms which is introduced in the picture below



In bonds language, in point 2 above we identify "satisfied" bonds around σ_I; the bond is satisfied if spins across the bond have the same orientation. Next, we "occupy" satisfied bonds with the probability P_{add}, add spins on occupied bonds into to the cluster, and compose a new list of satisfied bonds for just added spins (bonds may be considered for becoming occupied only once), etc.

The algorithm is ergodic because there is finite probability of flipping just one spin, and a sequence of such updates is sufficient to get an arbitrary configuration. As usual, the balance Eq. must be satisfied. To write it we compute the probability of constructing and flipping cluster $\mathcal{C}$ built from spin σ_I in configuration ν, and balance it with the backwards move of constructing and flipping exactly same cluster $\mathcal{C}$ built from the same initial spin σ_I in the configuration ν'. Since all spins in $\mathcal{C}$ are pointing in the same direction, the probability of occupying bonds between the cluster spins are identical for the direct and reversed updates. Also, the interaction energy between the cluster

spins is the same.



The energy difference between ν and ν' is only due to the "surface" of the cluster because ν and ν' differ by the number of bonds with are satisfied across the boundary of $\mathcal{C}$. Let us call the corresponding numbers K_ν and $K_{\nu'}$. Note, that $\mathcal{C}$-boundary bonds satisfied in ν are unsatisfied in ν' and vise versa, i.e. $K_\nu + K_{\nu'}$ account for all boundary bonds. The probability of not occupying K_ν bonds is $(1-P_{add})^{K_\nu}$—this is necessary for having $\mathcal{C}$ the way it is. The detailed balance Eq. then reads

$$
\begin{aligned}
&(1-P_{add})^{K_\nu}\, P_{acc}^{(\nu\to\nu')}\, e^{-\beta E_\nu} = (1-P_{add})^{K_{\nu'}}\, P_{acc}^{(\nu'\to\nu)}\, e^{-\beta E_{\nu'}} \\
&\frac{P_{acc}^{(\nu\to\nu')}}{P_{acc}^{(\nu'\to\nu)}} = (1-P_{add})^{K_{\nu'}-K_\nu}\, e^{\beta 2J(K_{\nu'}-K_\nu)} \\
&R = \left(\frac{1-P_{add}}{e^{-2J\beta}}\right)^{K_{\nu'}-K_\nu} . \qquad (4)
\end{aligned}
$$

It turns out that for $P_{add} = 1 - e^{-2J\beta}$ we get $R = 1$ and every update is accepted!

At high temperature $P_{add} \to 0$ and we are likely to perform a single spin-flip update, just like in the ordinary Metropolis algorithm. At low temperature, $P_{add} \to 1$ and we typically end up with the cluster containing most spins in the lattice. Not nice, but we already discussed how to handle the low-T phase properly.

The remarkable feature of the Wolff algorithm is that it virtually eliminates the critical slowing down problem by reducing the dynamical critical exponent z to very small value $z \leq 0.25$ [in fact, it is hard to tell whether z is finite or not, since the data for $t_0(L)$ can also be fitted to $A(1+\alpha \ln L)$

with $\alpha \approx 0.25$]. This means an enormous efficiency gain over the spin-flip schemes for the study of critical phenomena.

In the original Swendsen-Wang algorithm the whole lattice is partitioned into clusters in a single update. Namely, after the construction of $\mathcal{C} = \mathcal{C}_1$ is over, we may select any of the remaining spins at random and start building another cluster $\mathcal{C}_2$, excluding all bonds linking it to $\mathcal{C}_1$ from consideration. Then we build $\mathcal{C}_3$, etc. until we run out of spins.

In practice, we simply address all satisfied bonds in the lattice, occupy them with probabilities P_{add}, and identify clusters as spins connected by chains of occupied bonds. At this point, if we select at random some spin σ_I and flip the cluster it belongs to, we would get the Wolff algorithm because we repeated all steps of this algorithms exactly (we also did extra work by partitioning the rest of the lattice, but decided to "throw" it away). The Swendsen-Wang algorithm instead decides to flip each cluster with probability 1/2. The proof of detailed balance is the same as for the Wolff's version—only cluster boundary bonds are treated differently when proposing $\nu \to \nu'$ and going back $\nu' \to \nu$ but the probabilities of occupying such bonds are exactly compensated by the corresponding Gibbs factors.

Now, why Wolff's version of 1989 is better then the original 1987 scheme? From the discussion in previous paragraph we see that the probability of selecting a cluster to flip in the Wolff algorithm is proportional to its size, i.e. the algorithm is oriented towards flipping larger clusters more frequently. Flipping small clusters hardly makes sense because of the extra CPU time spent on the construction process which involves a lot of $rndm()$ calls, and our knowledge that single-spin updates are not efficient at the transition point. The difference between the Wolff and Swendsen-Wang approaches is not dramatic though.

In 1988 Niedermayer proposed a cluster scheme which allows to occupy all bonds, no matter satisfied or not. In his implementation, the probability of occupying the bond is a function of the bond state, e.g. its energy, $P_{add} = P_{add}(E_{<ij>})$, where $E_{<ij>} = -J\sigma_i \cdot \sigma_j$. For the Ising model satisfied bonds have $E_{<ij>} = -J$, and unsatisfied bonds have $E_{<ij>} = +J$. The algorithm itself is exactly the same as before with the only difference that all bonds are now considered for being occupied (but only once!). The balance Eq. is slightly different with respect to the bonds on the perimeter of the cluster (bonds between cluster spins do not change their states when the cluster is

flipped). Using previously introduced notations:

$$[1-P_{add}(-J)]^{K_\nu}\,[1-P_{add}(J)]^{K_{\nu'}}\,P_{acc}^{(\nu\to\nu')}\,e^{-\beta E_\nu} = \\ [1-P_{add}(-J)]^{K_{\nu'}}\,[1-P_{add}(J)]^{K_\nu}P_{acc}^{(\nu'\to\nu)}\,e^{-\beta E_{\nu'}}\ . \qquad (5)$$

The product of the two first factors is the probability of not occupying any of the cluster perimeter bonds. The acceptance ratio

$$R=\left(\frac{1-P_{add}(-J)}{1-P_{add}(J)}\,e^{2\beta J}\right)^{K_{\nu'}-K_\nu}\ , \qquad (6)$$

can be made perfect, i.e. ≥ 1, in a number of ways. $P_{add}(-J)=1-e^{-2\beta J}$; $P_{add}(J)=0$ is the Swendsen-Wang solution, but there are other possibilities too. For example,

$$P_{add}(E)=\begin{cases} 1-e^{\beta(E-E_0)} & \text{if } E<E_0 \\ 0 & \text{otherwise} \end{cases} \qquad (7)$$

with arbitrary $E_0 \geq J$ will result in $R \geq 1$. Niedermayer argued that free parameter E_0 may be used to further optimize the performance of cluster schemes, but this still remains a poorly researched proposal. For $E_0 < J$ the acceptance ratio is less than unity, and, finally, for $E_0 < -J$ the algorithm is reduced to the single-flip update because $P_{add}(E) \equiv 0$.

Invaded cluster algorithm (ICA) for critical points

This algorithm, invented by J. Machta et al. (1995), is an interesting deviation from the mainstream of MC methods because it is <u>not</u> based on the balance Eq. (!!!), and thus statistics of generated configurations is not representing the state of thermal equilibrium. However, it does find the critical point of the model automatically, and this is its main utility.

Let us first discuss differences between the low- and high-temperature states of the system in terms of cluster decomposition. In the disordered phase the spin-spin correlation function decays to zero at large distances, while in the ordered phase $g(i\to\infty)\to const$. We thus expect all cluster sizes much smaller than the system size at high T, and the largest cluster spreading through the whole system at low T. Using previously derived expression for magnetic susceptibility $\chi=\langle M^2\rangle/TN=\sum_i g(i)/T$ we immediately see that $\chi\to\infty$ when $T\to T_c$ (in the thermodynamic limit). In the

Swendsen-Wang algorithm all spins are partitioned into clusters which are then independently assigned random spin orientations. If n_I is the number of spins in cluster I, then $M = \sum_I s_I n_I$, and

$$\langle M^2 \rangle = \langle \sum_I s_I^2 n_I^2 \rangle + \langle \sum_{I \neq I'} s_I s_{I'} n_I n_{I'} \rangle = \langle \sum_I n_I^2 \rangle \tag{8}$$

The last equality follows from the fact that cluster orientations $s_I = \pm 1$ are random and uncorrelated. For χ to diverge at the critical point, it is crucial to have clusters with macroscopic sizes, i.e. $n_I \sim N^\alpha$ with positive α (at low T spins are strongly correlated and cluster sizes grow as large as $\sim N$).

Macroscopic clusters stretch across the entire system and it is possible to connect system boundaries by a network of occupied bonds. Such clusters are called "percolating". The outcome of this lengthy discussion is the relation between the percolating network just appearing in the system and the critical point. ICA uses this relation to determine T_c. It works as follows:

1. For any configuration of spins determine satisfied bonds and generate one random number for each satisfied bond.

2. By occupying bonds having their random numbers smaller than some initial parameter $p = p_{ini}$ we build clusters with sizes $n_I(p)$. If none of them is percolating, we increase p and keep growing clusters until one of them is percolating. The final value of p is denoted as p_{fin}. Formally, this procedure is identical to the conventional Swendsen-Wang update at temperature

$$T = -\frac{2J}{\ln(1 - p_{fin})} , \tag{9}$$

because p_{fin} is the probability of occupying a satisfied bond *by construction.*

3. Flip clusters as in the Swendsen-Wang algorithm and start over again from point 1.

Clearly, ICA is driving the system to the critical point by iteratively adjusting system temperature to be at the percolating point. In every update, system temperature is calculated from the probability of occupying satisfied bonds, Eq. (9). The outcome of the simulation is the average over the ICA process: $T_c(L) = \langle T \rangle_{ICA}$. As usual, the thermodynamic-limit answer is obtained by extrapolating a series of finite-size simulations $T_c = \lim_{L \to \infty} T_c(L)$.

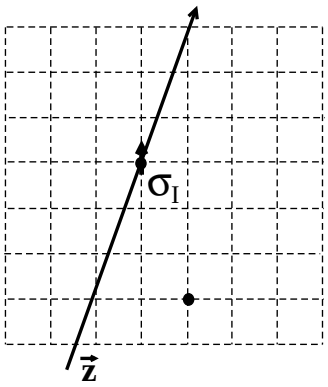
One may also study other system properties at the critical point such as $\chi(L)$ and $E(L)$.

The most important feature of ICA is its fast equilibration time. In most tests it equilibrates just in few MC updates, and its autocorrelation time is close to unity. This property is extremely important for multiprocessor simulations when MC statistics is increased by simply summing results of independent runs. When the thermalization time is short, very little CPU time is wasted on the initial thermalization process. Unfortunately, ICA works by constantly changing system temperature and thus properties of statistical fluctuations at the critical point (and even finite size corrections) can not be studied by this method. However, one may always "switch" from the ICA to the Swendsen-Wang mode in the same code and perform conventional stat. mech simulation at a fixed temperature.

Problem. Let some MC code with the autocorrelation time t_0 and thermalization time $t_{th} \gg t_0$ is used in the multiprocessor simulation (the number of processors is $M \gg 1$). The goal is to generate $K \gg M$ statistically independent configurations. Each processor runs independently. Estimate the required CPU time per processor. When the problem of long thermalization time is a serious issue?

Concluding remarks: All cluster methods discussed above can be easily reformulated for many other classical models, e.g. Potts, XY and O(N) systems. Consider, for example, how Wolff algorithm can be implemented for the continuous XY-model.

1. Choose at random the seed spin, σ_I, and the direction in the XY plane, $\vec{z}$. For every spin in the lattice the notion of "up-" or "down-spin" is defined with respect to the $\vec{z}$-axis, i.e. if $\vec{\sigma}_j \cdot \vec{z} > 0$ we say that $\vec{\sigma}_j$ is pointing up. Correspondingly, the bond is satisfied if σ_i and σ_j are both up- or down-spins; otherwise the bond is unsatisfied.

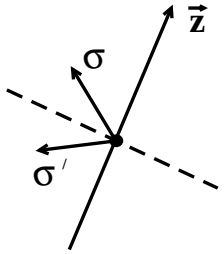


2. The rest of the update is done very much the same way as for the Ising model: consider all satisfied bonds around σ_I and occupy them with the probability

$$P_{add}(\vec{\sigma}_i, \vec{\sigma}_j) = 1 - e^{-2\beta J \sigma_i^z \sigma_j^z} \,, \quad \text{where} \quad \sigma^z = \vec{\sigma}\vec{z} \,. \tag{10}$$

For each new spin added to the cluster by occupying the bond, make a list of new satisfied bonds not considered before, etc ... The cluster construction continues until the list of new satisfied bonds is empty.

3. Flip the cluster with respect to the plane perpendicular to the $\vec{z}$-axis, i.e. change the sign of the spin projection on $\vec{z}$-axis, while keeping perpendicular components intact: $\vec{\sigma}' \equiv \vec{\sigma} - 2\vec{z}\sigma^z$.



We can always write the scalar product of n.n. spins as $\vec{\sigma}_i \cdot \vec{\sigma}_j = \sigma_i^z \sigma_j^z + \vec{\sigma}_i^{\perp} \cdot \vec{\sigma}_j^{\perp}$ by separating components $\parallel$ and $\perp$ to the $\vec{z}$-axis. The system energy

then takes the form

$$E = -J \sum_{<ij>} \sigma_i^z \sigma_j^z - J \sum_{<ij>} \vec{\sigma}_i^{\perp} \cdot \vec{\sigma}_j^{\perp} \ .$$

The second term remains constant in the update and is the same for ν and ν'. The first term is identical to the Ising model. We immediately realize that the algorithm has acceptance ratio $R \geq 1$ because it reproduces precisely what is done for the Ising model. The only extra feature (familiar from the Niedermayer's scheme) is that P_{add} is a function of the $\vec{z}$-axis contribution to the bond energy.

The efficiency of cluster methods is based on the possibility of having $R \geq 1$ for flipping arbitrary large clusters. Perfect acceptance ratio is obtained by clever building rules which take care of the surface energy. Since all spins in the cluster are flipped together, there is no energy penalty coming from the cluster bulk. However, this is true only in zero magnetic field, or, more generally, when the system Hamiltonian is invariant under flipping (rotating by the same angle) all spins. In finite magnetic the acceptance ratio gets an extra factor

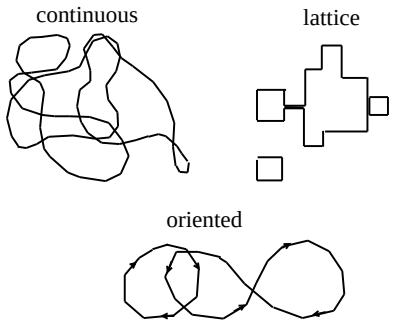
$$R(H \neq 0) = R(H = 0) \exp \left\{ -\frac{1}{T} \sum_{cluster} \vec{H} \cdot (\vec{\sigma}_i - \vec{\sigma}_i{}') \right\} \ ,$$

which is exponentially small for large clusters. Small acceptance ratio is obviously a very serious problem since most of the CPU time is wasted for nothing.

Classical Worm algorithms (WA)

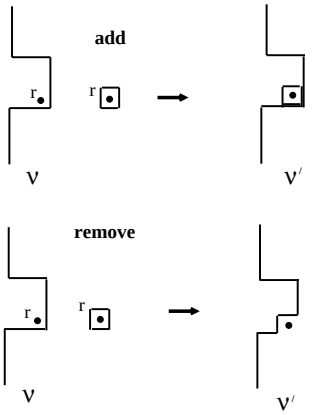
WA was originally introduced for quantum statistical models by Prokof'ev, Svistunov and Tupitsyn (1997), and later generalized to classical models by Prokof'ev and Svistunov (2001). The idea of WA is extremely simple. Imagine that the configuration space of the model can be represented by a collection of closed paths.

Many loops are allowed, they may overlap and intersect. In other words, anything you may draw with only one restriction—no open ends. In some models the path is oriented, i.e. one has to specify the direction along the line. Notice, that intersections make the "reading of the graph", i.e. deciphering how it was drawn, ambiguous, and this ambiguity is different for oriented and unoriented graphs.



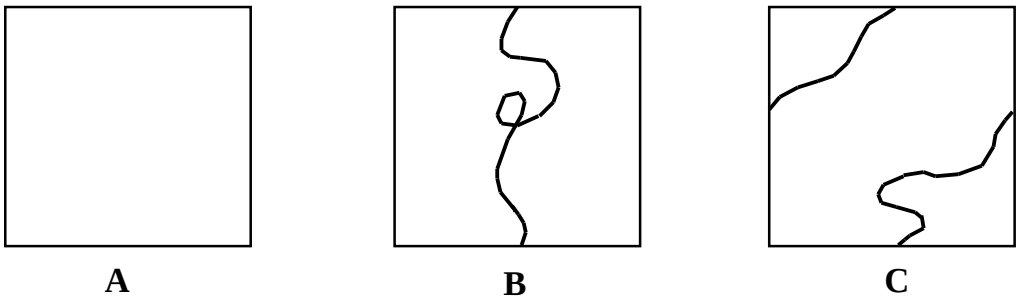
We will see shortly how closed path (CP) representation is obtained for the Ising, XY, and other classical models, but for now take it for granted. System energy is a function of the path, E_{path}, and the partition function is obtained by summing over all possible CP-configurations $Z = \sum_{\rm CP}\ e^{-E_{path}}$.

The most straightforward update of the CP-configuration is to add/remove an elementary loop (or plaquette, in lattice models). Only a small fragment of the CP configuration is shown here.



By repeating this procedure many times one can create paths of arbitrary

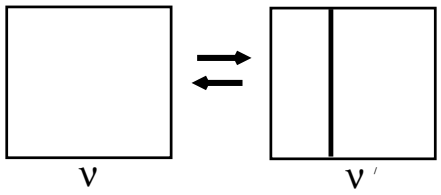
shapes and change the number of loops. One thing, however, can not be changed by adding and removing elementary loops, namely, loops which wind around the system with periodic boundary conditions. For example, in the figure below



A cannot be transformed to B and B can not be transformed to C (periodic boundary conditions are important here (!), otherwise the path in B and C is not closed).

To account for loops winding around the system one may supplement elementary-loop updates with global updates which propose to add/remove a large loop stretching across the system.

Global updates, however, have exponentially small acceptance ratio to insert the line, and exponential small probability to find one if we want to remove it, i.e. they are inefficient in large scale simulations.

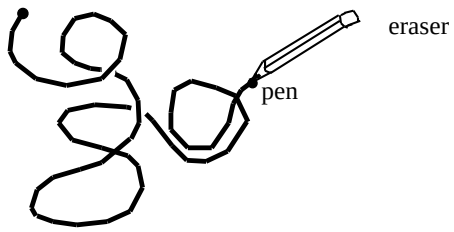


Worm Algorithm (WA) uses another strategy for updating CP-configurations. It consists of two major steps:

1. Configuration space is enlarged to include one disconnected loop, as if someone started drawing a new loop but is not finished yet. Let us denote such configurations as CP_g. From time to time the two ends of the disconnected loop meet at some point, and we recover the CP-configuration. In most cases configurations with one disconnected loop are not just an intermediate stage of the "drawing algorithm" trick, but represent important correlation functions, e.g. $g(i) = \langle \sigma_i \sigma_j \rangle$ (the derivation of this statement,

and more examples are given below).

2. All updates on CP_g configurations are performed **exclusively** through the end-points of the disconnected loop, no elementary-loop updates, no global updates. In essence, WA is literally a "draw" and "erase" procedure, exactly as you would produce a subtle tangle of loops on a paper.

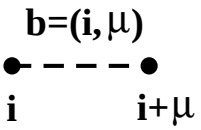


WA is a local Metropolis scheme, but, remarkably, its efficiency is similar to the best cluster methods at the critical point, i.e. it does not suffer from the critical slowing down problem. It has no problem producing loops winding around the system and transforming A to B and further to C. In what follows we will study how WA works for the Ising, XY, and $|\psi|^4$-models.

High-temperature expansions for the Ising model:
It seems, at first glance, that the Ising model has nothing to do with the CP configuration space. The trick is to switch from site to bond variables. Let us write the partition function as

$$Z = \sum_{\{\sigma_i\}} e^{-H/T} \equiv \sum_{\{\sigma_i\}} \prod_{b=(i,\mu)} e^{K\sigma_i\sigma_{i+\mu}} \ , \quad (K = \beta J) \ , \tag{11}$$

where b is the bond index; we label bonds by site i and direction $\mu = 1, 2, ..., d$, so that site $i + \mu$ is the n.n. of site i in direction μ. The convention is that $i - \mu$ is the n.n. of site i in the direction opposite to μ.



Before summing over spin variables we first expand exponents for each bond into Taylor series and rearrange terms

$$Z = \sum_{\{\sigma_i\}} \prod_{b=(i,\mu)} \left(\sum_{n_b=0}^{\infty} \frac{K^{n_b}}{n_b!} (\sigma_i\sigma_{i+\mu})^{n_b} \right) \equiv \sum_{\{n_b\}} \left(\prod_b \frac{K^{n_b}}{n_b!} \right) \sum_{\{\sigma_i\}} \left(\prod_i \sigma_i^{p_i} \right) \ , \tag{12}$$

where $p_i = \sum_\mu (n_{(i,\mu)} + n_{(i,-\mu)})$ is the sum of all "bond variables", n_b, for bonds connecting site i to its n.n. Summation over spins is trivial now because

$$\sum_{\sigma_i = \pm 1} \sigma^{p_i} = \begin{cases} 2 & \text{for even } \ p_i \\ 0 & \text{for odd } \ p_i \end{cases}$$

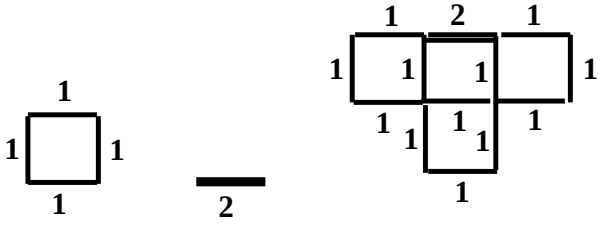
We are left with the sum over all possible bond variables which satisfy the constraint that on all sites p_i are even numbers

$$Z = 2^N \sum_{\{n_b\}_{CP}} W(\{n_b\}) \ ; \qquad W(\{n_b\}) = \prod_b \frac{K^{n_b}}{n_b!} \ . \tag{13}$$

As usual, W is the weight of the bond configuration defined as a collection of $\{n_b\}$ numbers.

If we present graphically each term in the expansion over K by a line on the corresponding bond ($K^{n_b}/n_b!$ is presented by n_b lines), then the $\{p_i\}$=even constraint is equivalent to the condition that bond lines form a closed path configuration. Indeed, if we draw a continuous closed path we always enter and exit each site and ensure that lines are added to the site in pairs.

A typical $\{n_b\}_{CP}$ configuration is shown in the figure, with bond variables mentioned next to the solid line.



So far we did not make any progress in the solution of the Ising model; we simply formulated it in terms of statistics of un-oriented loops.

Next, consider the spin-spin correlation function

$$g(j_2 - j_1) = Z^{-1} \sum_{\{\sigma_i\}} \sigma_{j_1} \sigma_{j_2} \, e^{-H/T} \equiv G(j_2 - j_1)/Z \ . \tag{14}$$

We may now repeat all steps of the expansion in terms of bond variables and perform summation over spins for G. The only difference between G and Z is that for two special sites, j_1 and j_2, we now have $\sigma^{p_{j_1}+1}$ and $\sigma^{p_{j_2}+1}$.

As a consequence, bond configurations with non-zero weight must satisfy a constraint: $\{p_i\}$ are all even with the exception of sites j_1 and j_2 where p_{j_1} and p_{j_2} are odd unless $j_1 = j_2$. Graphically, sites j_1 and j_2 are the end-points of some disconnected path, i.e. configurations contributing to G are the CP_g-configurations representing intermediate stages of the "drawing" process. Thus we have

$$g(i) = \frac{\sum_{\{n_b\}_{CP_g}} W(\{n_b\})\, A_{j_1, j_2 = j_1 + i}}{\sum_{\{n_b\}_{CP}} W(\{n_b\})} \,, \tag{15}$$

where A is simply a "reminder" telling us the distance between the end-points, or the relative location of σ_{j_1} and σ_{j_2} variables in the correlation function.

What we did above is also known as a high-temperature expansion because $K = \beta J \to 0$ as $T \to \infty$ and the expansion is converging fast. It is almost self-evident now that this expansion is perfectly suited for WA. The algorithm itself has only one **"shift"** update which straightforwardly implements the draw-and-erase procedure. The convention is to call the end points "ira"=j_1-site and "masha"=j_2-site.

1. If $ira = masha$, select at random a new site j and assign $ira = masha = j$; otherwise skip this step. It is introduced to be able to start drawing a new line from any site when the current path is closed. Select at random the direction (bond) to shift masha to the n.n. site, let it be j_3, and whether this shift will increase or decrease the bond number by one, i.e. whether we would like to draw or erase the line.

2. Perform the shift if $rndm() < R$: assign $masha = j_3$, and undate the corresponding bond number $n_b = n_b \pm 1$. This is it!

The acceptance ratio is given by the ratio of the configuration weights:

$$R = \frac{2d}{2d} \begin{cases} K/(n_b+1) & \text{when the bond number is increased} \\ n_b/K & \text{when the bond number is decreased} \end{cases} \tag{16}$$

In this algorithm every configuration is contributing to the spin-spin correlation function; if $i = masha - ira$ then you may add to the statistics of $G(i) = G(i) + 1$. The partition function is updated $Z = Z + 1$ only when $ira = masha$, i.e. when the configuration is CP. This correlation function

is automatically normalized properly because we get $g(0) = 1$ as expected, $\langle\, \sigma_i\sigma_i\rangle = \langle\, 1\rangle = 1$. Magnetic susceptibility is just the sum over all points for the correlation function $\chi = \sum_i g(i)/T$. As usual, the estimator for energy is obtained from $E = -\partial \ln Z/\partial\beta$, but now we have to use the representation (13) to apply it:

$$E = -Z^{-1} \sum_{CP} \sum_b (n_b/\beta)\, W(\{n_b\}) = -T\langle\, N_b\rangle\ , \tag{17}$$

where $N_b = \sum_b n_b$ is the sum of bond numbers in the configuration. The corresponding counter is easy to keep track of; when the "shift" update is accepted, $N_b = N_b \pm 1$.

Problem. Figure out how to calculate the average energy through the correlation function, i.e. without using Eq. (17).

There are a couple of notes worth discussing here. It went unmentioned in the discussion above that CP and CP_g configurations can, in general, have arbitrary relative weights even when $ira = masha$. This arbitrary relative weight, let us call it ω_G, is removed at the end of the calculation by normalizing the correlation function to the known answer, for example, if it is known that $g(0) = C$, then we apply the following transformation to the MC data $g(i) = C[g(i)/g(0)]$. For the Ising model we are lucky that $\sigma_i^2 = 1$, and thus $C = 1$; next, we discuss the $|\psi|^4$-model where $C \neq 0$. The other observation is that in the update shifting masha to the ira-site we do not have the $1/N$ factor in the acceptance ratio. Formally, the probability of going backwards is $1/2dN$, but we can always assume that when shifting masha to the ira-site we also go from CP_g to the CP configuration and thus their relative weight ω_G should be mentioned in the balance equation (below $masha' = ira$)

$$P_{acc}^{(masha\to masha')}\, \omega_G\, (1/2d)\, W_\nu = P_{acc}^{(masha'\to masha)}\, (1/2dN)\, W_{\nu'}\ . \tag{18}$$

By choosing $\omega_G = 1/N$ we compensate the $1/N$ factor on the r.h.s. and obtain properly normalized $g(i)$. Of course, it makes sense to select ω_G so that the normalization of the correlation function is automatic, but, unfortunately, sometimes the true value of $g(0)$ is known only from the same MC simulation. In any case, Eq. (18) may be used instead of Eq. (16) if necessary.

The other note is specific to the Ising model. Instead of expanding bond exponentials into the Taylor series we can use the following identity

$$e^{K\sigma_i\sigma_j} \equiv \cosh(K)(1+\tanh(K)\,\sigma_i\sigma_j) \equiv \cosh(K)\sum_{n_b=0}^{1}\tanh^{n_b}(K)(\sigma_i\sigma_j)^{n_b}\,,$$

This simple identity is possible because the $\sigma_i\sigma_j$ product has only **two values**, ± 1. This representation has an advantage that bond numbers n_b can take only two values 0 and 1, i.e. we have a smaller configuration space—this makes a big difference in the algorithm performance at low temperature when $K \to \infty$ and $\tanh(K) \to 1$. The rest in the algorithm is exactly as before with $R = \tanh^{\pm 1}(K)$.

Problem. Write down an explicit expression for the partition function for this $n_b = 0, 1$ representation, and use it to derive the estimator for energy,

$$E = -J\tanh(K)\,[2N + \langle\, N_b\rangle/\sinh^2(K)]\,.$$

Problem. This is a long one. Develop a WA code with the shift update for the Ising model and use it to calculate the spin-spin correlation function in 2D. Take $K = K_c$ and $L^2 = 100^2$. Do not forget to thermalize for 10^8 updates before you start. Ask me for tricks if necessary.

High-temperature expansion and WA for the $|\psi|^4$ model:
The model itself is defined by the Hamiltonian (i, j and k below are d-dimensional vectors)

$$H = -t\sum_{<ij>}[\psi_i^*\psi_j + \psi_i\psi_j^*] - c\sum_i|\psi_i|^2 + (U/2)\sum_i|\psi_i|^4\,, \qquad (19)$$

where $\{\psi_i\}$ are complex number variables. The first term cam be written in the diagonal form

$$H_k = \sum_k \epsilon_k|\psi_k|^2\,, \quad \epsilon_k = -2t\sum_\mu\cos(k\mu)\,. \qquad (20)$$

if we use the Fourier representation for ψ_i

$$\psi_k = N^{-1/2} \sum_j e^{-ikj} \psi_j \ ,$$

$$\psi_j = N^{-1/2} \sum_k e^{ikj} \psi_k \ .$$

The second and third terms describe linear and nonlinear interactions between the complex fields. I will skip here arguments how this model can be relevant for simulations of the weakly-interacting Bose gas, and simply note that it can be obtained from the quantum Hamiltonian

$$H = -t \sum_{<ij>} [b_i^\dagger b_j + b_i b_j^\dagger] - c \sum_i n_i + (U/2) \sum_i n_i^2 \ , \tag{21}$$

where b_i and $b_i^\dagger$ are bosonic creation and annihilation operators, and $n_i = b_i^\dagger b_i$ are site occupation numbers. The first term describes particle hopping between lattice sites, the second term is then the chemical potential contribution (often considered formally as an integral part of the Hamiltonian in the Grand canonical ensemble simulations), and, finally, U describes on-site interactions between particles. This model is also known as lattice **Bose Hubbard model**. If we replace $b_i \to \psi_i$ we will transform (21) to (19). This transformation does not change the physics of the model only under special conditions (not discussed here), but we will keep notations and constants names as they were for the quantum case, e.g. $n_i = |\psi_i|^2$ and $n_k = |\psi_k|^2$ are occupation numbers in the site and momentum representations, and $N_p = \sum_i n_i = \sum_k n_k$ is the total number of "particles".

Problem. Explain the connection between the XY and $|\psi|^4$ models. More specifically, show that the XY-model is a specific limiting case of the $|\psi|^4$ model.

The partition function of the $|\psi_i|^4$ model reads

$$Z = \prod_i \left(\int d\psi_i \right) e^{-H/T} \ .$$

It depends on the ratios of $t/T = \tilde{t}$, $c/T = \tilde{c}$ and $U/T = \tilde{U}$. In fact, the dependence on one ratio is trivial and can be singled out by rescaling

fields. Since fields ψ are continuous and their real and imaginary parts vary from $-\infty$ to ∞ we may apply the scaling transformation $\psi \to \bar{\psi}/\lambda$ without changing the structure of the partition function integral. Then

$$Z = \lambda^{2dN} \prod_i \left(\int d\bar{\psi}_i \right) e^{\tilde{S}} , \tag{22}$$

where

$$S = -H/T = \frac{t}{T\lambda} \sum_{<ij>} [\bar{\psi}_i^* \bar{\psi}_j + h.c.] - \frac{c}{T\lambda} \sum_i |\bar{\psi}_i|^2 + \frac{U}{2T\lambda^2} \sum_i |\bar{\psi}_i|^4 .$$

One may choose $\lambda = t/T$ to make the coefficient in front of the first term equal to unity and write $Z = (T/t)^{2dN} f(\bar{c}, \bar{U})$, where $\bar{c} = c/t$, and $\bar{U} = UT/t^2$. In what follows I will not use this property, though.

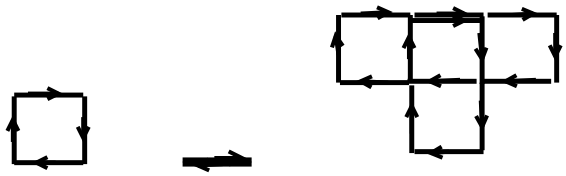
The high temperature expansion proceeds as follows. Expand bond exponentials into Taylor series

$$e^{(t/T)\psi_i^* \psi_j} = \sum_{n=0}^{\infty} (\psi_i^* \psi_j)^n \, (t/T)^n / n! ,$$

$$e^{(t/T)\psi_i \psi_j^*} = \sum_{m=0}^{\infty} (\psi_i^* \psi_j)^m \, (t/T)^m / m! .$$

Now we have to exponentials for the bond: one corresponds to the particle hopping from site j to site i and the other describes hopping events in the opposite direction. Graphically, we may use directed lines to specify terms in each expansion (line arrows specify the direction of the hopping event).

Now we have two exponentials for the bond: one corresponds to the particle hopping from site j to site i and the other describes hopping events in the opposite direction. Graphically, we may use directed lines to specify terms in each expansion (line arrows specify the direction of the hopping event). It is also convenient to call directed lines "currents".



The obvious convention is that for bond between n.n. sites i and j the incoming current for i is the outgoing current for j.

The partition function takes the form

$$Z = \sum_{\{n_b,m_b\}} \left(\prod_b \frac{(t/T)^{n_b+m_b}}{n_b! m_b!} \right) \prod_i \left(\int d\psi_i \, \psi_i^{p_i} (\psi_i^*)^{r_i} e^{-(U/2T)|\psi_i|^4 + (c/T)|\psi_i|^2} \right) , \tag{23}$$

where $p_i = \sum_\mu n_{(i,\mu)} + n_{(i,-\mu)}$ and $r_i = \sum_\mu m_{(i+\mu,-\mu)} + m_{(i-\mu,\mu)}$ are the sums over all outgoing and incoming currents correspondingly. When the integral over the complex field is written in the polar or (module,angle) coordinates

$$\int d\psi \psi^p (\psi^*)^r e^{-(U/2T)|\psi|^4 + (c/T)|\psi|^2} = \int d|\psi| |\psi|^{p+r+1} e^{-(U/2T)|\psi|^4 + (c/T)|\psi|^2} d\varphi e^{i\varphi(p-r)} ,$$

it becomes obvious that due to the phase integral the non-zero result is obtained only for $p = r$, i.e. when the number of incoming and outgoing currents are equal for each site. Once again, this is the CP constraint on the allowed bond or current configurations. The remaining integral is a function of the site "charge" p

$$Q(p) = \int dx \, x^{2p+1} e^{-(U/2T)x^4 + (c/T)x^2}$$

(recall that for the Ising model the corresponding site factors were all the same, $Q = 2$). These integrals are not a big deal in terms of practical implementation of the code, since they can be easily tabulated (for a given set of Hamiltonian parameters) at the beginning of the MC run.

Our problem is now formulated in terms of summation over directed CP-configurations

$$Z = \sum_{\{n_b,m_b\}_{CP}} W(\{n_b, m_b\}) , \quad W(\{n_b, m_b\}) = \prod_b \left(\frac{(t/T)^{n_b+m_b}}{n_b! m_b!} \right) \prod_i Q(p_i) . \tag{24}$$

Similarly to the Ising model, the correlation function $g(i = j_2 - j_1) = \langle \psi^*_{j_1} \psi_{j_2} \rangle \equiv G(i)/Z$, also known as the **Green function**, is described by the CP_g-configurations for its numerator. Extra $\psi^*_{j_2}$ and ψ_{j_1} factors require that on the corresponding sites $p - r = \pm 1$, i.e. one extra current line is entering/exiting the site.

At this point we recover the familiar set up suitable for the WA "draw and erase" procedures. Since the configuration weight now depends explicitly on the location of end-points even when $ira = masha$ it makes sense to consider the procedure of taking the pen out of the paper and placing it somewhere

else (without drawing any lines) as a separate **"relocate" update**. Thus is $ira = masha = j_0$ we may choose with probability p_{re} to do the "relocate" update by selecting at random a new site j and, if $j \neq j_0$ and $rndm() < R$, assigning new values $ira = masha = j$. The acceptance ratio for this simple update is

$$R = \frac{Q(p_{j_0})\, Q(p_j + 1)}{Q(p_{j_0} + 1)\, Q(p_j)}$$

The **"shift" update** is almost identical to the shift update for the Ising model:

1. Select at random the direction (bond) to shift $masha$ to the n.n. site, let it be j_3, and whether this shift will increase the outgoing current of decrease the incoming current by one, i.e. whether we would like to draw the outgoing line or erase the incoming line. This update is accepted with probability $p_{sh} = 1 - p_{re}$ if $ira = masha$ and probability $p_{sh} = 1$ if $ira \neq masha$. Correspondingly, the shift update in the backwards direction is selected with probability $p'_{sh} = 1 - p_{re}$ if $ira = j_3$ and $p_{sh}{}' = 1$ if $ira \neq j_3$

2. Perform the shift if $rndm() < R$: assign $masha' = j_3$, and undate the corresponding bond, $n_b = n_b + 1$ or $m_b = m_b - 1$, and site numbers.

The acceptance ratio is given by the ratio of configuration weights and probabilities to select the shift procedure

$$R = \frac{2d\, p_{sh}{}'}{2d\, p_{sh}} \begin{cases} \frac{t}{T(n_b+1)}\, \frac{Q(r_{j_3}+1)}{Q(r_{j_3})} & \text{when the outgoing current number is increased} \\ \\ \frac{Tm_b}{t}\, \frac{Q(r_j-1)}{Q(r_j)} & \text{when the incoming current number is decreased} \end{cases} \tag{25}$$

Since configurations are generated with probabilities proportional to their contributions to the G-function, the statistics for the correlation function can be updated after every MC step, $G(masha - ira) = G(masha - ira) + 1$. When $ira = masha$ the configuration of bond numbers is of the CP-type, but its weight in the $G(i = 0)$-sector is by a factor $Q(p_{ira} + 1)/Q(p_{ira})$ different from its weight in the Z-sector since it contains an extra $|\psi_{ira}|^2$ factor. Thus the MC estimator for the partition function is obtained by removing (in our mind) one extra $|\psi_{ira}|^2$ factor

$$Z = Z + Q(p_{ira})/Q(p_{ira} + 1)\ .$$

The average particle density $n = \langle\, |\psi_i|^2 \rangle$ is at the same time the normalization factor for the correlation function $n = g(0) = G(0)/Z$. The average interaction energy is given by $\langle\, |\psi_{ira}|^4 \rangle$ which has one more $|\psi_{ira}|^2$ factor on site ira. Its estimator is then $Q(p_{ira}+2)/Q(p_{ira}+1)$ when $ira = masha$. The hopping energy is given by the sum of bond numbers $-t\langle \sum_b (n_b + m_b)\rangle$; this formula derives from the now familiar approach $\langle H_k \rangle = -t\partial \ln Z/\partial t$.

Superfluid stiffness and winding numbers.

This section is not about efficient MC methods. It discusses how one can study the system response to the gauge field—the most important question in the theory of superfluidity. We are still looking at the properties of classical systems and one may object that superfluidity is a quantum phenomenon. In short, the answer is NO, it is not quantum as far as the general principles leading to superfluidity are concerned. Superfluidity arises when there is a well defined, single valued complex order parameter field, $\Psi = |\Psi| e^{\phi}$. True, often it originates from the quantum system wavefunction, but complex numbers can be considered as vectors in the XY-plane and classical systems such as XY- and $|\psi|^4$-models have similar macroscopic properties.

I will use the lattice $|\psi|^4$-model with periodic boundary conditions to introduce the gauge field and twisted boundary conditions, but same considerations apply to continuous quantum/classical systems. Lattice gauge field is defined by phases added to the hopping amplitudes

$$-t \sum_{<ij>} \psi_i^* \psi_j + c.c. \longrightarrow - \sum_{<ij>} (t\, e^{i\phi_{<ij>}}) \psi_i^* \psi_j + c.c.\ . \tag{26}$$

Since ψ_i are complex numbers, it does not change anything in the system if we substitute $\psi_i \to e^{i\delta_i}\, \psi_i$. This will, however, result in the **gauge transformation** from $\phi_{<ij>}$ to $\tilde{\phi}_{<ij>} = \phi_{<ij>} - \delta_i + \delta_j$. All gauge transformed fields are equivalent to each other. If by proper choice of δ_i we can make $\tilde{\phi}$ field equal to zero (or multiple of 2π) everywhere, then it has no effect on the system.

In continuous models, gauge fields are introduced by making long space derivatives

$$\nabla \longrightarrow [\nabla - i\mathbf{a}]\ . \tag{27}$$

The connection between lattice gauge fields and more familiar continuous version is obtained by considering slowly varying ψ and ϕ fields, or the limit of small lattice constant l, and identifying $\phi_{i,\mu} = -l\mathbf{a}_\mu(i)$ (using site-direction

index to specify the bond). Notice, that in this limit $\phi_{i,\mu} \to 0$. Then, writing Eq. (26) explicitly in the self-conjugated form

$$-t \sum_{i\mu} \left(\psi_i^* \psi_{i+l\mu}\, e^{-il\mathbf{a}_\mu(i)} + \psi_i^* \psi_{i-l\mu}\, e^{il\mathbf{a}_\mu(i-\mu)} \right) , \tag{28}$$

and expanding to order l^2 one can show that it reduces to

Problem.

$$-\frac{1}{2m} \sum_i \psi_i^* \, [\nabla - i\mathbf{a}]^2 \psi_i \; .$$

where $1/2m = tl^2$. Do the derivation and get this result from Eq. (28).

We see that lattice gauge theories are more general than continuous ones because the latter represent only a particular limiting case of the former. Thus, all results derived for the lattice gauge field automatically apply to continuous systems as well.

Consider now a gauge field which has non-zero bond phases only along one direction, let it be the $\hat{x}$-direction [recall also that i is the d-dimensional index $i = (x, y, \ldots)$]

$$\phi_{i,\mu} = \delta_{\mu,\hat{x}} f(x)/L_x \; , \tag{29}$$

where $f(x)$ is arbitrary and L_x is the system size in $\hat{x}$-direction. We observe that only the closed loop sum over all sites in the selected direction, $\sum_x f(x) \equiv \Phi$, is physically meaningful; indeed, under gauge transformation $\delta_{x+l\hat{x}} = \delta_x - f(x)$, all bond phases $\tilde{\phi}_{i,\mu}$ become zero except $\tilde{\phi}_{L_x,\hat{x}} = \Phi$. So far, dealing with periodic boundary conditions, we insisted that sites obtained by going L_μ steps in the direction μ are identical to themselves, i.e. $\psi_{i+L_\mu\mu} \equiv \psi_i$. This is exactly why we are not capable of eliminating all bond phases by the gauge transformation performed on the ψ variables. If instead we formally allow

$$\psi_{i+L_\mu\mu} = e^{-i\Phi} \psi_i \; , \tag{30}$$

then all gauge phases become zero but our ϕ variables are not single valued anymore unless Φ is a multiple of 2π. Eq. (30) is known under the name of **twisted boundary conditions**. They are nothing but another way of looking at things which may be useful in some calculations. Twisted boundary conditions, are automatically satisfied if we, for simplicity, "spread out"

the gauge phase Φ evenly between the x-bonds; in other words, we add a constant gradient term to the phase of the ψ_i field: $\psi_i \to e^{i\varphi(x)}\psi_i$, where $\varphi(x) = \Phi x/L_x$. For fixed Φ and large L_x the added phase gradient is macroscopically small, $\nabla\varphi(x) = \Phi/L_x \to 0$.

Next, we look at the system response to this gauge field, or twisted boundary conditions. The partition function, and all other thermodynamic quantities are now functions of Φ. In finite-size systems the dependence on Φ is analytic, and for small Φ one may use Taylor series to write the free-energy F as

$$F(\Phi) - F(0) = \frac{\rho_s^{(x)}\Phi^2}{2m}\,\frac{V}{L_x^2}\ . \tag{31}$$

This equation may be considered as the definition of ρ_s—it is a linear response coefficient describing free-energy change in response to the gauge field phase. Depending on the context, it is called the **superfluid density**, or the **helicity modulus**, or the **spin stiffness**. The name does not matter much here. The linear in Φ term is absent, because H is a real function (hermitian operator in quantum case), and thus is an even function of phase: $H(-\Phi) = H^*(\Phi) = H(\Phi)$. Correspondingly, Z and other thermodynamic functions are also even in Φ. In a more general treatment, the gauge field is non-zero in all directions, $\Phi = (\Phi_x, \Phi_y, \ldots)$, and the superfluid density is a tensor,

$$F(\Phi) - F(0) = \left(\frac{\rho_s^{(x)}}{2m}\right)_{\mu\nu}\,\frac{V\Phi_\mu\Phi_\nu}{L_\mu L_\nu}\ .$$

For simplicity, we will consider below only isotropic systems with $(\rho_s/m)_{\mu\nu} = \rho_s/m\,\delta_{\mu\nu}$ and equal linear sizes in all directions $L_\mu = L$.

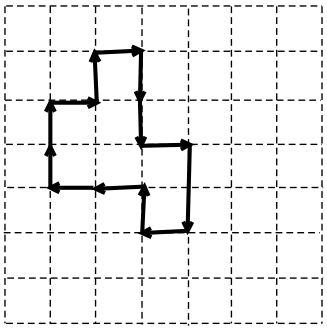
What about system size dependent factors in (31)? They are introduced to ensure that $(\rho_s/2m)(\nabla\varphi)^2$ has the meaning of the free-energy density which is system size independent, and $(\rho_s/m)\nabla\varphi$ has the meaning of the current density. The above expression is then a volume integral $F = \int dV(\rho_s/2m)(\nabla\varphi)^2$.

Since ρ_s is a macroscopic response to the external perturbation, it has no microscopic definition in terms of ψ_i variables. It turns out, however, that in the high-temperature expansion approach and its CP-configuration space of bond currents used for WA, the $Z(\Phi)$ dependence can be easily determined from the statistics of oriented loops winding around the system in space direction. First, we notice that if all configurations of oriented lines are closed loops then the sum of all currents going through any plane

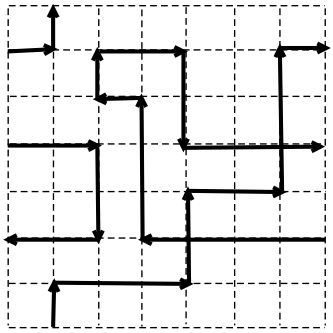
perpendicular to the x-direction is the same

$$M_x(R) = \sum_{b=(i,\hat{x})} (n_b - m_b)\delta_{i_x,R} = const \equiv L_x^{-1} \sum_{b=(i,\hat{x})} (n_b - m_b) \; . \qquad (32)$$

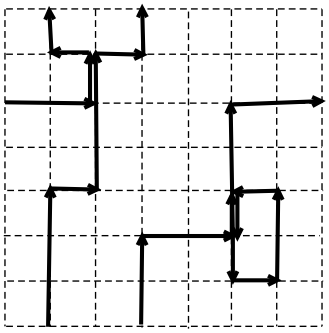
otherwise there are more lines entering the region between two planes than exiting them and thus there are end points in this region. This property can be used to define the **winding number** in the x-direction as the sum over all bond-currents in this direction divided by L_x, see the last identity in Eq. (32). M_x is an integer. It is called a winding number, because loops which *do not* wind around the system and *can be* reduced to small size using local transformations of its shape have zero contribution to winding numbers. It means that configurations with different winding numbers are topologically distinguishable, i.e. they may not be transformed into each other using local (small in size) modifications of their shape. Some examples are shown in the picture below.



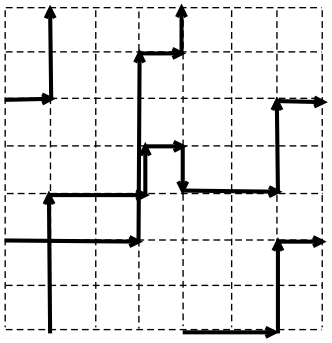
$M_x = 0$



$M_x = 1$



$M_x = 1$



$M_x = 2$

The second figure can be decomposed into two loops in a number of ways,

but one of the loops is always winding around the system, while the other can be reduced to zero by small changes of its shape. Of course, winding numbers, can be introduced to characterize topological properties of loops in all d directions, $\mathbf{M} = (M_x, M_y, \ldots)$.

Let us now examine what would happen to the configuration contribution to Z when the gauge field is applied. Since $W(\{n_b, m_b\})$ depends on the hopping amplitude as, see Eq. (24), $\sim t^{\sum_b n_b} t^{\sum_b m_b}$, in the presence of the gauge field it will transform as

$$W(\Phi) = W(0)\, e^{i(\Phi_\mu/L)\sum_{b=i,\mu} n_b}\, e^{-i(\Phi_\mu/L)\sum_{b=i,\mu} m_b} = W\, e^{i\Phi\cdot\mathbf{M}} ,$$

(recall that the n_b lines originate from $-t\psi_i^*\psi_j$ terms and m_b lines originate from complex conjugated ones, that is why the phase factors have opposite signs for n_b and m_b lines). Correspondingly, the partition function is

$$Z(\Phi) = \sum_{\mathbf{M}} Z_{\mathbf{M}}(\Phi = 0)\, e^{i\Phi\cdot\mathbf{M}} , \tag{33}$$

where $Z_{\mathbf{M}}$ is the partition function of all configurations having the same winding number. The ratio $Z_{\mathbf{M}}/Z(0)$ is the probability of finding the equilibrium state with the winding number $\mathbf{M}$.

The rest is pure math.

$$F = -T \ln Z ,$$

$$\frac{\partial F}{\partial \Phi_x} = \frac{-T}{Z(\Phi)} \sum_{\mathbf{M}} iM_x Z_{\mathbf{M}}\, e^{\Phi\cdot\mathbf{M}} ,$$

$$\left.\frac{\partial^2 F}{\partial \Phi_x^2}\right|_{\Phi=0} = T\left\{\frac{\sum_{\mathbf{M}} M_x^2 Z_{\mathbf{M}}}{Z} - \left(\frac{\sum_{\mathbf{M}} M_x Z_{\mathbf{M}}}{Z}\right)\left(\frac{\sum_{\mathbf{M}'} M_x' Z_{\mathbf{M}}}{Z}\right)\right\} = T\langle M_x^2\rangle ,$$

Since $(\partial F/\partial\Phi_x)|_{\Phi_x=0} = 0$ (no linear in Φ terms), we also have $\langle M_x\rangle = 0$. Our final relation between $\rho_s = 2mL^{2-d}\partial^2 F/\partial\Phi_x^2|_{\Phi=0}$ and the winding number statistics is then

$$\rho_s = 2mTL^{2-d}\langle M_x^2\rangle \equiv \frac{2mT}{d}\, L^{2-d}\, \langle \mathbf{M}^2\rangle , \tag{34}$$

where I made use of $\langle \mathbf{M}^2\rangle = d\langle M_x\rangle$ in the isotropic system.

The possibility of collecting winding number histograms, $Z_{\mathbf{M}}/Z$, is extremely important in studies of superfluid systems, and is considered as an advantage of WA over cluster methods. It is also about the adequate model

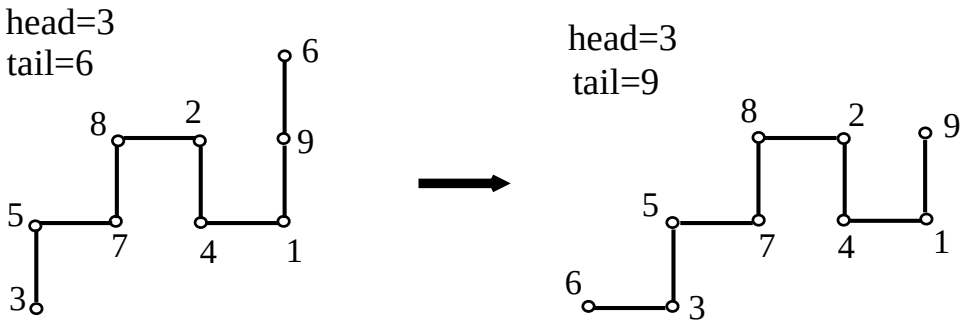
representation since winding numbers are not defined in the original ψ_i-field representation.

WA for polymers:

In the previous Section we discussed the snake algorithm for polymers which translates the chain along itself by one step. If the polymer chain is consisting of **identical monomers** then the following two modifications make the snake-WA algorithm a much better numerical tool capable of simulating efficiently both dilute and dense polymer systems.

1. When monomers are identical, we may describe the polymer as a collection of L points (not necessarily ordered, so that R_i and R_{i+1} may be far away from each other) with associations between them specified by the $ass_{\pm}(i)$-arrays: if $j = ass_{\pm}(i)$ then site $R(j)$ is the nearest neighbor of $R(i)$. Most associations do not change in the standard snake move. For example, when the "head" (head and tail are special points with only one association) is shifted from R to the n.n. point R_{new} we simply change associations for the polymer end points:

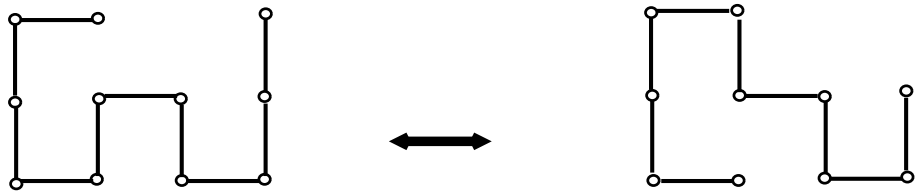
newhead=tail
tail=ass$_+$(tail)
ass$_+$(head)=newhead
ass$_-$(newhead)=head
head=newhead
R(head)=R$_{new}$



This update is local and its computational cost does not scale with L any more.

2. WA enters the picture by adding one more update which reconnects

the polymer line (that is why having all monomers the same). When the new head point R_{new} happens to be one of the polymer points, we may choose to reconnect it as shown in the picture below.



This update *reconnects* the polymer chain locally, and its acceptance ratio does not contain macroscopic factors. However, reconnections change the global topology of the chain, tight and relax topological knots, and allow the head to move easily through the maze of monomers in the dense (globule) phase. And there is a price to be paid for that—in the reconnection procedure all associations along the loop between the old and new head positions must be reversed (updated)— on average $\sim L$ operations.

Flat histogram or entropic sampling. Wang-Landau approach.

Cluster methods and WA solve the problem of critical slowing down for continuous transitions, but they are of little help in the study of first-order phase transitions and glassy systems. An efficient algorithm for these type of systems should abandon the idea of generating configurations with probabilities proportional to their weights and use another principle which allows easy transitions between energy minima separated by large barriers. There are several methods which achieve this goal.

The idea of **entropic sampling** is to generate system states with probabilities resulting in the flat energy distribution histogram, i.e. in this method the probability of having a state with energy E is energy independent. Formally,

$$Z = \sum_{\nu} e^{-E_\nu/t} \equiv \sum_{E} \left(\sum_{\nu} \delta_{E_\nu,E} \right) e^{-E/T} = \sum_{E} \rho(E)\, e^{-E/T} \; .$$

The density of states can also be written as an exponential $\rho(E) = e^{S(E)}$, where $S(E)$ is the entropy. Then,

$$Z = \sum_E \, e^{-(E-TS(E))/T} \equiv \sum_E \, e^{-F(E)/T} \; .$$

When states are generated with probabilities proportional to Gibbs weights $\sim e^{-E_\nu/T}$, the energy histogram is given by $e^{-\beta[F(E)-F(\langle\, E\rangle)]}$, where $\langle\, E\rangle$ is the point of the free-energy minimum. To have a flat distribution we have to generate states with energies away from $\langle\, E\rangle$ with probabilities increased by a factor $e^{\beta[F(E)-F(\langle\, E\rangle)]}$, i.e. with probability (note that constant factors like $\langle\, E\rangle$ or $F(\langle\, E\rangle)$ can be absorbed into the distribution normalization and thus can be added or dropped at will)

$$\sim e^{-\beta[E-\langle\, E\rangle]} \, e^{\beta[F(E)-F(\langle\, E\rangle)]} \sim e^{-S(E)} \; .$$

The last relation explains why the method is called entopic sampling. Now, instead of Gibbs factors, $e^{-E_\nu/T}$, we have to use entropy exponentials $e^{-S(E_\nu)} = 1/\rho(E_\nu)$ in the balance equation. For example, the acceptance ratio for the single spin-flip update in the Ising model will become $R = \rho(E_\nu)/\rho(E_{\nu'})$. Correspondingly, all averages will transform to (what we did is no different from the standard reweighing technique)

$$\langle\, A\rangle_{MC} = \frac{\sum_\nu A_\nu \rho(E_\nu) e^{-E_\nu/T}}{\sum_\nu \rho(E_\nu) e^{-E_\nu/T}} \equiv \frac{\sum_\nu A_\nu e^{-F(E_\nu)/T}}{\sum_\nu e^{-F(E_\nu)/T}} \; .$$

There are two problems with the entropic sampling. One is that the exponential $e^{-F(E_\nu)/T}$ may overflow. We already know how to handle the overflow problem by writing everything in terms of logarithms. The other issue is that $\rho(E)$ is not known at the beginning of MC simulation! Entropy is a statistical property; it is *not* a function of a given configuration. It can be simulated, though, by MC methods. The idea then is to start entropic sampling with some initial, rough approximation of what $\rho(E)$ might be. Let us denote the initial guess as $\rho_0(E)$. From reweighing technique we know that final answers do not depend on the probability distribution used to generate configurations (only efficiency does), thus in the long run it does not matter what ρ_0 is.

First, we study the density of states itself by collecting energy histogram, $h_0(E_i)$ =how many times we had $E_\nu \in (E_i - \Delta E/2, E_i + \Delta E/2)$, (here $\{E_i\}$

is the set of histogram points). Recall, that our goal is to have it flat. The density of states is then

$$\rho_1(E_i) = h_0(E_i)\,\rho_0(E_i)\ .$$

Since the initial guess ρ_0 is not perfect, and statistics is exponential in E, it is unlikely that $h_0(E)$ will be nearly constant. Most probably, it will be peaked in one region and have zeros in other histogram bins. Correspondingly, $\rho_1(E_i)$ is also far from being the final result for the density of states (we actually have no idea what $\rho_1(E_i)$ is if the i-th bin has no entries), though it is a better one than ρ_0. To avoid zero values for ρ_1 we can use $\rho_1(E_i) = \rho_0(E_i)$ if $h_0(E_i) = 0$. At this point one may substitute ρ_0 with ρ_1 and start collecting a new energy histogram h_1. It is used then to improve the knowledge of the density of states for the next iteration

$$\rho_{n+1}(E_i) = \begin{cases} h_n(E_i)\,\rho_n(E_i)\ , & \text{if } \ h_n(E_i) \neq 0 \\ \rho_n(E_i)\ , & \text{if } \ h_n(E_i) = 0 \end{cases}$$

The procedure is repeated many times until the histogram h_n is nearly flat. Then the actual MC simulation starts.

It is hard to tell how long it takes to achieve the flat histogram goal because reweighing factors are enormous $\sim e^N$ exponentials and we have to sample all scales with reliable statistics. On the other hand, if $\rho(E)$ is known, we may calculate system properties at all temperatures at once! For example

$$\langle\, E\rangle = \frac{\sum_E \rho(E) E e^{-E/T}}{\sum_E \rho(E) e^{-E/T}}\ .$$

Or, if at the final stage of the calculation one splits the statistics of A_ν into energy bins,

$$A(E) = \frac{\sum_\nu A_\nu \delta_{E_\nu \in i-bin}}{\sum_\nu \delta_{E_\nu \in i-bin}} = \langle\, A\rangle_{over\ i-bin}\ ,$$

then

$$\langle\, A\rangle = \frac{\sum_E \rho(E) A(E) e^{-E/T}}{\sum_E \rho(E) e^{-E/T}}\ .$$

In both cases temperature shows up only in the final analysis when the MC calculation is already done.

Of course, nothing is for free. It takes much longer CPU to have similar errorbars in the entropic sampling method than in the conventional scheme

with T kept fixed. After all, in the last equation only a small region of energies around $F(E)$ minimum is contributing to the answer, and these energies constitute a small fraction $\sim 1/\sqrt{N}$ of the total MC statistics.

Recently, Landau & Wang (2002) found a clever way of implementing the procedure looking for the flat histogram distribution. Instead of improving the knowledge of $\rho(E)$ iteration after iteration with relatively large number of MC sweeps per iteration, they proposed to modify $\rho(E)$ currently used by the algorithm after every update! In the standard approach it is a very slow process of expanding the region of energies visited by the simulation because the initial guess is not good. Landau & Wang suggested a procedure which *forces* the simulation to explore energies not visited before and thus quickly accumulate points all over the histogram.

The implementation is very simple. Let the current density of states function used in the acceptance ratio is $\rho_{MC}(E)$. The initial guess can be anything, e.g. $\rho_{MC}(E) = 1$. After each MC update, whatever is the current configuration energy, E_ν, one takes the corresponding value of $\rho_{MC}(E_\nu)$ and multiplies it by a modification factor, $f > 1$

$$\rho_{MC}(E_\nu) \to \rho_{MC}(E_\nu)\, f \ . \tag{35}$$

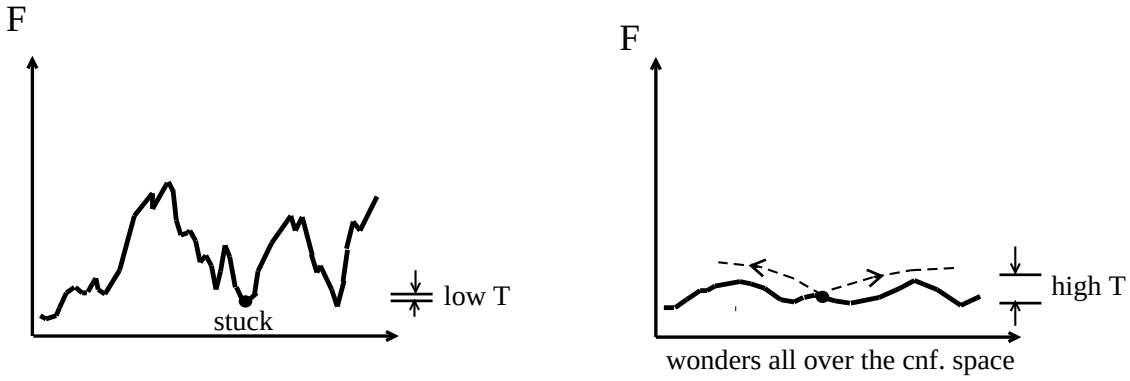
The initial choice of f is formally arbitrary, but limited not to be very large to ensure that the scheme remains stable. This is all! Of course, we also collect the energy histogram. If the histogram is "reasonably" flat, i.e. flat with certain accuracy, the calculation is stopped. In the next round, the modification factor f is decreased to a smaller value (the suggested law is $f \to \sqrt{f}$), and the simulation continues with the energy histogram being reset. Iteration after iteration, the modification factor is decreased and is made very close to unity, e.g. $f = 1.00000001$ or similar. Since at final stages the modification factor is hardly changing $\rho_{MC}(E)$ and the histogram is flat, we conclude that the scheme has converged to the true density of states.

So, why does it work better than the original scheme? Since the acceptance ratio is given by $R(E \to E') = \rho_{MC}(E)/\rho_{MC}(E')$, the increase of $\rho_{MC}(E)$ by a factor of f increases R by a factor of f too, and even if the acceptance ratio is initially small it is bound to grow exponentially until we accept the move, etc. The algorithm is literally forcing the configuration to change its energy state. If the histogram is checked for flatness every K sweeps then a good the rule of thumb for selecting the initial value of f (according to Landau and Wang) is $f^K \sim \Upsilon$, where Υ is the total number of

states in the system. I understand this rule as an insurance policy that f^K can beat any mistake in the initial density of states because $\rho(E) < \Upsilon$, by definition.

Simulated and parallel tempering techniques for glasses

This method was proposed by Marinari and Parisi in 1992 for spin glasses. Overcoming large energy barriers is painful at low temperature. At high temperature, especially if it is above the glass transition point T_g, transforming system configurations is relatively easy. The idea then is to make temperature itself one of the system parameters which can be changed in updates, and make sure that the histogram $h(T_i)$, i.e. how long the systems is spending at temperature T_i, is nearly flat. The last requirement is necessary to ensure that both the high-T and low-T states are sampled. Literally, in the simulated tempering, we heat and cool the system while keeping all configurations at any given T in the state of thermal equilibrium. This allows to simulate the low-T configuration space more efficiently when there are multiple valleys in the free-energy landscape separated by barriers.



Similar to the entropic sampling many temperatures are simulated at once and there should be some problem with severe reweighing in the method. Suppose we do simulated tempering using Metropolis algorithm as it is. The only extra feature is an update suggesting to change the temperature from T_i to T_{i+1} or T_{i-1} by ΔT. We decide whether T_i is increased or decreased with probabilities $p_i^{(\pm)}$, and can use $p_i^{(\pm)} = 1/2$ for all values of i except for i_{max} (then $p_i^{(+)} = 0$ and $p_i^{(-)} = 1$) and i_{min} (then $p_i^{(+)} = 1$ and $p_i^{(-)} = 0$). Since only temperature, but not the energy is changed, the acceptance ratio

is

$$R = \exp\left\{-E_\nu\left(\frac{1}{T_{i\pm1}} - \frac{1}{T_i}\right)\right\} \frac{p_{i\pm1}^{(\mp)}}{p_i^{(\pm)}} \sim \exp\left\{\pm\frac{\Delta T E_\nu}{T_{i\pm1}T_i}\right\} . \qquad (36)$$

First, we notice that the temperature change ΔT should be relatively small, $\Delta T \sim 1/N$ to ensure that acceptance ratios are of order unity in both directions. This is a hurdle, because to get from T_{min} to T_{max} we need $\sim N$ temperature points and it will take $\sim N^2$ steps or more to make the round-trip journey in temperature. Second, we notice that the histogram $h_i \equiv h(T_i)$ will not be flat unless we use reweighing. The probabilities of being at different temperatures in this algorithm relate as combined weights of all configurations, i.e. as their partition functions. To make the histogram flat we have to multiply the acceptance ratio by $Z(T_i)/Z(T_{i\pm1})$. This is not a problem, except that $Z_i/Z_{i\pm1}$ are not known prior to simulation, and we have to apply an iterative, self-adaptive procedure (e.g. Landau-Wang style) to find it in the first place. When the partition function ratios are known to certain degree of accuracy, the simulation may go. There is, however, another elegant trick which entirely eliminates this problem.

Parallel tempering consumes more memory but this is not an issue for glasses where system sizes are relatively small. Now, instead of making temperature a MC variable, we simply simulate in parallel $\sim K \gg 1$ replicas of exactly the same system, but at different temperatures T_i. Temperatures are never changed , and thus there is no reweighing problem. The new feature is the exchange of configurations between replicas at neighboring temperatures (we select the n.n. pair of replicas at random)

$$E_\nu^{(at\ T_i)},\ E_\mu^{(at\ T_{i\pm1})} \longrightarrow E_\mu^{(at\ T_i)},\ E_\nu^{(at\ T_{i\pm1})} ,$$

with the acceptance ratio

$$R = \exp\left\{\left(\frac{1}{T_i} - \frac{1}{T_{i\pm1}}\right)(E_\nu - E_\mu)\right\} = e^{\pm\Delta T\Delta E} . \qquad (37)$$

The second factor is nothing but the product of two Gibbs exponentials. The detailed balance is obviously satisfied because from the "perspective" of replica "i" we do the standard job

$$P_{acc}^{\nu\to\mu}\, e^{-E_\nu/T_i}\, e^{-E_\mu/T_{i\pm1}} = P_{acc}^{\mu\to\nu}\, e^{-E_\mu/T_i}\, e^{-E_\nu/T_{i\pm1}} ,$$

where the first exponential is the configuration weight and the second one is proportional to the probability of selecting the new configuration from the equilibrium set at another temperature.

We see that there is no need for histograms of h_i at all—just make sure that each replica is updated for $\sim t_0(i)$-sweeps and do $\sim K$ random configuration swaps then. Neighboring replicas frequently overlap in energies if $\sqrt{C}T \sim C\Delta T$, i.e. when

$$\Delta T/T \sim 1/\sqrt{C} \sim 1/\sqrt{N} \, ,$$

(recall that C is the specific heat). It means that $K \sim \sqrt{N}$ is sufficient to have T_{max} twice as high as T_{min}.