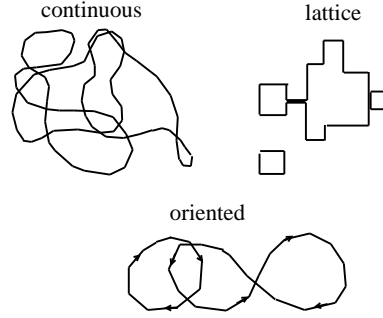


Classical Worm algorithms (WA)

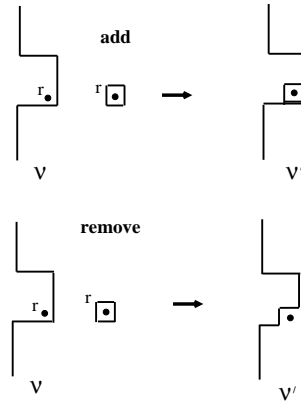
WA was originally introduced for quantum statistical models by Prokof'ev, Svistunov and Tupitsyn (1997), and later generalized to classical models by Prokof'ev and Svistunov (2001). The idea of WA is extremely simple. Imagine that the configuration space of the model can be represented by a collection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



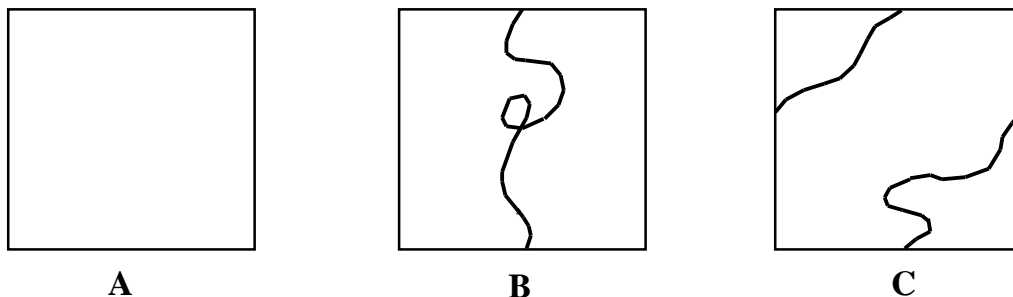
We will see shortly how closed path (CP) representation is obtained for the Ising model but for now take it for granted. System energy is a function of the path, E_{path} , and the partition function is obtained by summing over all possible CP-configurations $Z = \sum_{CP} e^{-E_{path}}$.

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

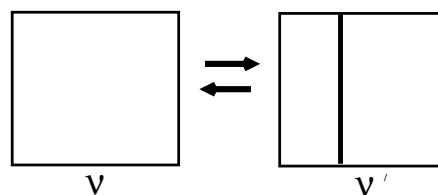
shapes and change the number of loops. One thing, however, can not be changed by adding and removing elementary loops, namely, loops which wind around the system with periodic boundary conditions. For example, in the figure below



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.

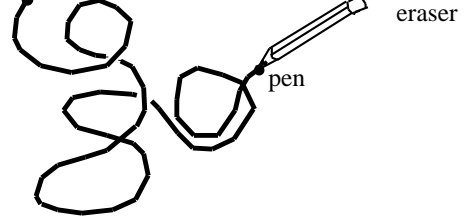


Worm Algorithm (WA) uses another strategy for updating CP-configurations. It consists of two major steps:

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 model.

High-temperature expansions for the Ising model:

Ising model describes a system of lattice spin variables coupled by the nearest neighbor (n.n.) interaction that is defined by the Hamiltonian

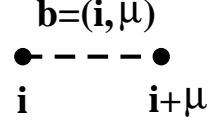
$$H = -J \sum_{\langle ij \rangle} \sigma_i \sigma_j. \quad (1)$$

Here J is the coupling energy, $\langle ij \rangle$ stands for the n.n. pairs, and $\sigma_i = \pm 1$.

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

$$Z = \sum_{\{\sigma_i\}} e^{-H/T} \equiv \sum_{\{\sigma_i\}} \prod_{b=(i,\mu)} e^{K \sigma_i \sigma_{i+\mu}}, \quad (K = \beta J), \quad (2)$$

where b is the bond index; we label bonds by site i and direction $\mu = 1, 2, \dots, d$, 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

$$Z = \sum_{\{\sigma_i\}} \prod_{b=(i,\mu)} \left(\sum_{n_b=0}^{\infty} \frac{K^{n_b}}{n_b!} (\sigma_i \sigma_{i+\mu})^{n_b} \right) \equiv \sum_{\{n_b\}} \left(\prod_b \frac{K^{n_b}}{n_b!} \right) \sum_{\{\sigma_i\}} \left(\prod_i \sigma_i^{p_i} \right), \quad (3)$$

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

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

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

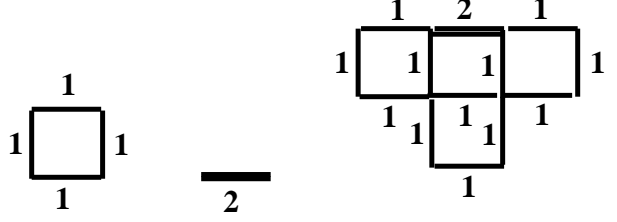
$$Z = 2^N \sum_{\{n_b\}_{CP}} W(\{n_b\}); \quad W(\{n_b\}) = \prod_b \frac{K^{n_b}}{n_b!}. \quad (4)$$

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 $\{p_i\}=\text{even}$ 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

$$g(j_2 - j_1) = Z^{-1} \sum_{\{\sigma_i\}} \sigma_{j_1} \sigma_{j_2} e^{-H/T} \equiv G(j_2 - j_1)/Z . \quad (5)$$

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

$$g(i) = \frac{\sum_{\{n_b\}_{CP_g}} W(\{n_b\}) A_{j_1, j_2=j_1+i}}{\sum_{\{n_b\}_{CP}} W(\{n_b\})} , \quad (6)$$

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.

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:

$$R = \frac{2d}{2d} \begin{cases} K/(n_b + 1) & \text{when the bond number is increased} \\ n_b/K & \text{when the bond number is decreased} \end{cases} \quad (7)$$

In this algorithm every configuration is contributing to the spin-spin correlation function; if $i = (\text{distance between } masha \text{ and } ira)$ 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 (4) to apply it:

$$E = -Z^{-1} \sum_{CP} \sum_b (n_b / \beta) W(\{n_b\}) = -T \langle N_b \rangle, \quad (8)$$

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, $N_b = N_b \pm 1$.

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 $masha' = ira$)

$$P_{acc}^{(masha \rightarrow masha')} \omega_G (1/2d) W_\nu = P_{acc}^{(masha' \rightarrow masha)} (1/2dN) W_{\nu'} . \quad (9)$$

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. (9) may be used instead of Eq. (7) 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 $K \rightarrow \infty$ and $\tanh(K) \rightarrow 1$. The rest in the algorithm is exactly as before with $R = \tanh^{\pm 1}(K)$, and the estimator for energy has the form

$$E = -J \tanh(K) [d N + \langle N_b \rangle / \sinh^2(K)] . \quad (10)$$