Monte Carlo simulation

- Partition function ($S \to H/T$ in statistical physics language)

$$Z = \int [\prod_x d\phi] e^{-S}$$

- We want to evaluate Z and *especially*

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int [d\phi] \mathcal{O} \, e^{-S}$$

- Well defined and in principle straightforward integral. However:

A) Dimensionality: (number of space-time points)×(dim. of fields)

$\approx N_S^3 \times N_T \times 4 \times 8$ (for SU(3))

$\sim 10^5 - 10^9$

$\to$ "regular" integration methods fail $\to$ Monte Carlo?

B) All states with equal S contribute with equal weight. However,

$$\int [d\phi] \, e^{-S} = \int dS n(S) \, e^{-S}$$

is an integral over sharply localized distribution. Only states with $S \lesssim \langle S \rangle$ contribute, which is typically a tiny fraction of the whole phase space. This kills std. Monte Carlo, which samples over whole domain (different versions ...).

- *Importance sampling:*

- Select N configurations ϕ_a, distributed according to the probability

$$p(\phi) = \frac{1}{Z} e^{-S}$$

- Measure observable A from these; $A_a = A[\phi_a]$. - Expectation value of A:

$$\bar{A} = \frac{1}{N} \sum_a A[\phi_a] \quad \to \quad \langle A \rangle \quad \text{when } N \to \infty$$

This follows from the probability distribution of A:

$$p(A) = \int [d\phi] p(\phi) \delta(A - A[\phi])$$
$$\rightarrow \bar{A} \equiv \int dA\, A\, p(A) = \int [d\phi]\, A[\phi] p(\phi) = \langle A \rangle \qquad (1)$$

- *How to generate configurations with $p(\phi) = \frac{1}{Z} e^{-S}$?*
- f is an "update", a modification of an existing configuration ϕ: $\phi \overset{f}{\mapsto} \phi'$. It can be here a "small" update, one d.o.f, or a "large" one, for example all d.o.f's (usually just repeated application of small updates).
- $P_f(\phi \mapsto \phi')$ is the probability distribution of the resulting configuration:

$$\int d\phi' P_f(\phi \mapsto \phi') = 1$$

- *Requirements:*

  A) f must preserve the equilibrium distribution $p_{\text{eq.}}(\phi)$:

$$\int d\phi P_f(\phi \mapsto \phi') e^{-S(\phi)} = e^{-S(\phi')}$$

  B) f must be *ergodic:* starting from *any* configuration, repeated application of f brings us to arbitrarily close to any other configuration.

- Sufficient to guarantee that *any* initial ensemble $\rightarrow p_{\text{eq.}}$ as f is applied to it ("thermalization").
- It also guarantees that the distribution of the configurations in a single Markov chain

$$\phi_0 \mapsto \phi_1 \mapsto \phi_2 \mapsto \phi_3 \ldots$$

  approaches $p_{\text{eq.}}$.

- Proof: let $p(\phi)$ be the probability distribution of an initial ensemble of configurations. After applying f once, we obtain distribution $p'(\phi)$.

$$\begin{aligned} |p' - p_{\text{eq.}}| &\equiv \int d\phi |p'(\phi) - p_{\text{eq.}}(\phi)| = \int d\phi | \int d\phi' P_f(\phi' \mapsto \phi)(p(\phi') - p_{\text{eq.}}(\phi'))| \\ &\leq \int d\phi \int d\phi' P_f(\phi' \mapsto \phi)|p(\phi') - p_{\text{eq.}}(\phi')| = |p - p_{\text{eq.}}| \end{aligned}$$

  If $P_f > 0$, then $\leq \rightarrow <$ unless $p = p_{\text{eq.}}$.

- All standard methods satisfy detailed balance:

$$\frac{P_f(\phi \mapsto \phi')}{P_f(\phi' \mapsto \phi)} = e^{S(\phi) - S(\phi')}$$

Common update methods:

- Let us consider the most common *local* update methods: 1) heat bath, 2) Metropolis and 3) overrelaxation. Illustrate with O(2) σ-model:

$$S = -\beta \sum_{<ij>} s_i \cdot s_j = -\beta \sum_{<ij>} \cos(\theta_i - \theta_j)$$

  Here s_i is 2d vector with $|s_i| = 1$.

- Choose a variable at site i for updating

- *local action:* $S_i(s_i) = -\beta \sum_{j=\text{n.n.of}\, i} s_i \cdot s_j = s_i \cdot V = v \cos\theta$

  where $v = |V|$ and θ is the angle between s_i and V.

- *Heat bath:*

  Choose new s_i with probability

$$p(s_i) = e^{-S_i(s_i)} \Big/ \int ds' e^{-S_i(s')}$$

  - Satisfies detailed balance

- Computer gives random numbers from uniform distribution, $X \in [0,1)$. We obtain s from $p(s)$ using

$$X = \int_{\min}^{s} ds' p(s')$$

- We're unlucky: S_i is not very easily integrable. Try with O(3).

- Integrability is often a problem. There are methods around it.

- Heat bath is common in gauge theories, efficient implementation ∃ for SU(2)

- *Metropolis:* (Metropolis *et al*, 1953)

  Choose new s_i' with

  $$s_i' = s_i + C(X - 0.5),$$

  C is a tunable constant. s_i' is *accepted* with probability

  $$p_M(s_i \mapsto s_i') = \min(1, e^{S_i(s_i) - S_i(s_i')}),$$

  if rejected, leave s_i as is.

  - Satisfies detailed balance.

  - More generally: we can choose s_i' using any distribution $f(s_i, s_i')$, which satisfies $f(a,b) = f(b,a)$.

  - Function f (or constant C) is tuned to optimize the performance; usually so that the acceptance is $\sim 50 - 70\%$.

  - Very versatile.

- *Overrelaxation:*

  Reflect s_i to the "other side" of the potential S_i:

  $\theta \to -\theta$, or equivalently $s_i \to 2(s_i \cdot V)v^{-2}V - s_i$.

  - Deterministic, very fast

  - Satisfies detailed balance

- Not ergodic: S never changes (*microcanonical*). Must be usually mixed with other updates.

- Often more efficient than the others above.

- If $S_i(s)$ is not symmetric, care has to be taken to ensure correctness (try).

- Not applicable to discrete spins.

Autocorrelations

- Let X_n is a measurement of some quantity from configuration number $n = 1 \ldots N$; At finite N, the estimated error:

$$\delta X = \sqrt{\frac{\sum_i (X_i - \langle X \rangle)^2}{N(N-1)}}$$

if and only if the measurements X_i are statistically independent.

- In reality (Markov chain), successive configurations are correlated, *autocorrelations*:

Autocorrelation function of X is

$$C(k) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N-k} X_n X_{n+k} \propto e^{-k/\tau}$$

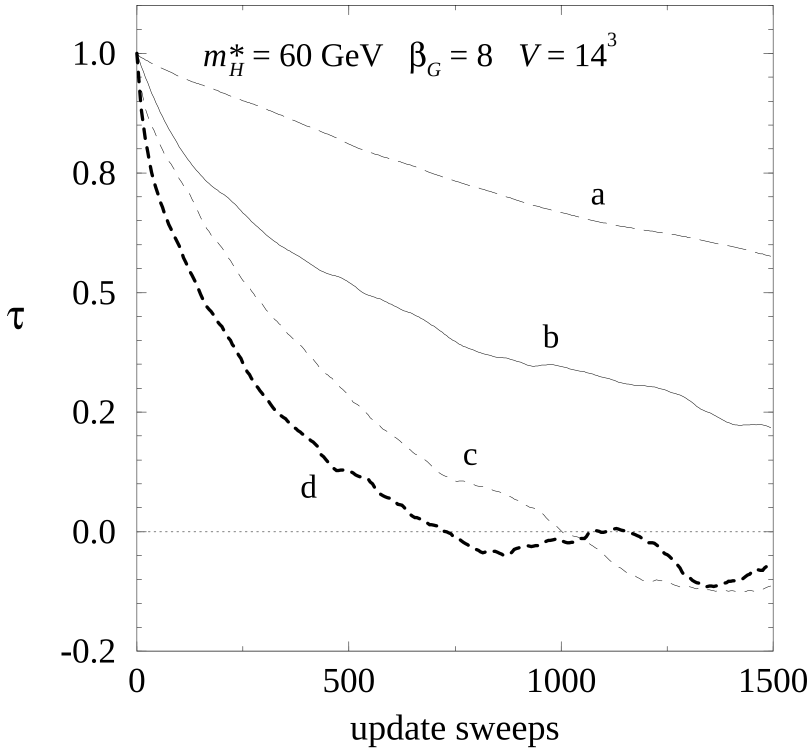
Here τ is the *autocorrelation time*.

- # of independent configs is $\sim N/(1+2\tau_a) \to \delta X \approx \sqrt{1+2\tau_a}\delta X_{\mathrm{naive}}$.

- Goal: invent update algorithms which minimize τ. Or rather, which minimize the ratio τ/(cpu-time) or τ/(your time).

- Often: *OR* is better than *HB* is better than *Metropolis*. There are exceptions. However, *cluster algorithms* are in a class of their own.

- Autocorrelation function measured from 3d SU(2) + Higgs theory:



(a) Heat bath + Metropolis

(b) Heat bath + Metropolis + global R update: $R \to \xi R$

(c) $4\times$ overrelaxation + $1\times$ (heat bath + global)

(d) $4\times$ ("wavefront overrelaxation) + $1\times$ (heat bath + global)

Cluster algorithms

Fundamentals: Fortuin-Kasteleyn cluster decomposition

Ising model (arbitrary dim.):

$$E = -\sum_{<i,j>} s_i s_j \qquad Z = \sum_{\{s\}} e^{-\beta E}\,.$$

($s_i = \pm 1$, $< i,j >$ nearest neighbour sites). Consider interaction between sites $< l,m >$, and remove it from E:

$$E_{l,m} = -\sum_{<i,j>\neq<l,m>} s_i s_j\,.$$

Define

$$Z^{\text{same}}_{l,m} \equiv \sum_{\{s\}} \delta_{s_i,s_j} e^{-\beta E_{l,m}}\,, \qquad Z^{\text{diff.}}_{l,m} \equiv \sum_{\{s\}} (1-\delta_{s_i,s_j}) e^{-\beta E_{l,m}}\,.$$

Now we can write $Z = e^{\beta} Z^{\text{same}}_{l,m} + e^{-\beta} Z^{\text{diff.}}_{l,m}$. Further, using

$$Z^{\text{ind.}}_{l,m} \equiv \sum_{\{s\}} e^{-\beta E_{l,m}} = Z^{\text{same}}_{l,m} + Z^{\text{diff.}}_{l,m}\,.$$

the partition function becomes

$$Z = (e^{\beta} - e^{-\beta}) Z^{\text{same}}_{l,m} + e^{-\beta} Z^{\text{ind.}}_{l,m}$$

Since the first term restricts $s_l = s_m$, and the second term contains no restriction, the weighting factors can be considered as relative probabilities of a bond between sites l,m and no bond (= independent states).
Normalizing: $p^{\text{same}} = 1 - e^{-2\beta}$, $p^{\text{ind.}} = e^{-2\beta}$.

Repeating for all $< i, j >$,

a) sites linked to each other by bonds form clusters

b) different clusters are independent: $s_c = \pm 1$

and the partition function can be written as

$$Z = \sum_{\text{bonds}} \sum_{s_c} p^b (1-p)^n = \sum_{\text{bonds}} p^b (1-p)^n 2^{N_c}$$

Here

$$\begin{aligned} p &= 1 - e^{-2\beta} \\ b &= \# \text{ of bonds}, \\ n &= \# \text{ of links that do not form a bond}, \\ N_c &= \# \text{ of clusters}. \end{aligned}$$

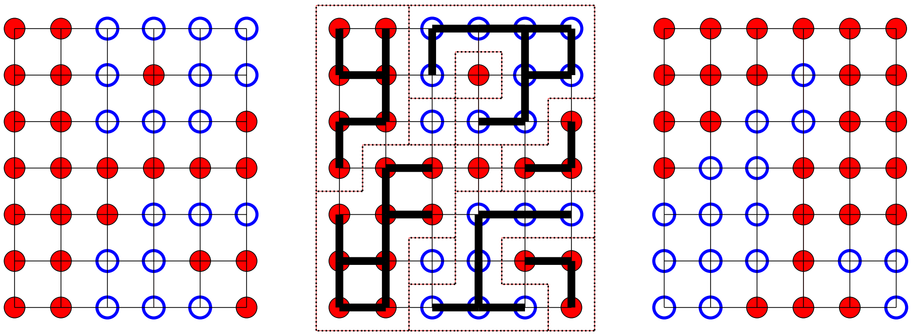
This decomposition is at the core of the cluster algorithms.

Swendsen-Wang cluster update

[Swendsen and Wang, PRL 58 (1987) 86]

Beginning with an arbitary configuration s_i, one SW cluster update cycle is:

1. Inspect all nn-states s_i, s_j. If $s_i = s_j$, create a bond between sites i, j with probability $p = 1 - \exp(-2\beta)$ (otherwise no bond).
2. Construct clusters = sets of points connected by bonds.
3. Set each cluster to a random value ± 1.



The configuration changes a lot in one update!
Is this a valid update? It must satisfy

a) ergodicity (obvious)

b) detailed balance: $\dfrac{P(A \mapsto B)}{P(B \mapsto A)} = \exp -\beta(E_B - E_A)$?

Proof: consider $A \mapsto \mathcal{C} \mapsto B$, where $\mathcal{C}$ is some bond configuration compatible with both A and B. Now $P(\mathcal{C} \mapsto A) = P(\mathcal{C} \mapsto B) = 1/2^{N_c}$, and

$$\frac{P(A \mapsto \mathcal{C})}{P(B \mapsto \mathcal{C})} = \frac{(1-p)^{d_A}}{(1-p)^{d_B}} = \exp -\beta(E_B - E_A)$$

where $d_{A,B}$ are the numbers of similar nn-states which are not connected by a bond. Thus $A \mapsto \mathcal{C} \mapsto B$ and $B \mapsto \mathcal{C} \mapsto A$ satisfy detailed balance for arbitrary $\mathcal{C}$, and the total transition probabilities $A \mapsto B$, $B \mapsto A$ must do it also.

Wolff single cluster update

[U. Wolff, PRL 62 (1989) 361]

Principle: do the cluster decomposition as in S-W, but invert ('flip') only one randomly chosen cluster! In practice:

1. Choose random site i.
2. Study neighbouring sites j. If $s_j = s_i$, join site j to cluster with probability $p = 1 - \exp(-2\beta)$.
3. Repeat step 2 for site j, if it was joined to the cluster. Keep on doing this as long as the cluster grows.
4. When the cluster is finished, invert the spins which belong to it.

• Usually slightly more effective than S-W (the average size of the clusters is larger. Why?).
• The minimum cluster size = 1, maximum = volume.
• Nicely recursive.
• Satisfies detailed balance.

As an example, let us consider 2-dimensional Ising model. It has 2nd order phase transition at $\beta = \log(1 + \sqrt{2}) \approx 0.44$ (Curie point).
Correlation length (and cluster size!) diverges when one approaches the critical point.

Compare the performance of
a) Metropolis and
b) Wolff cluster algorithm.

Cluster update in c-pseudocode:

Here is a c-code snippet which should perform Wolff cluster update. `s[i]`: spin array, and `dran()` is a random number generator which returns a number fro 0 to 1.

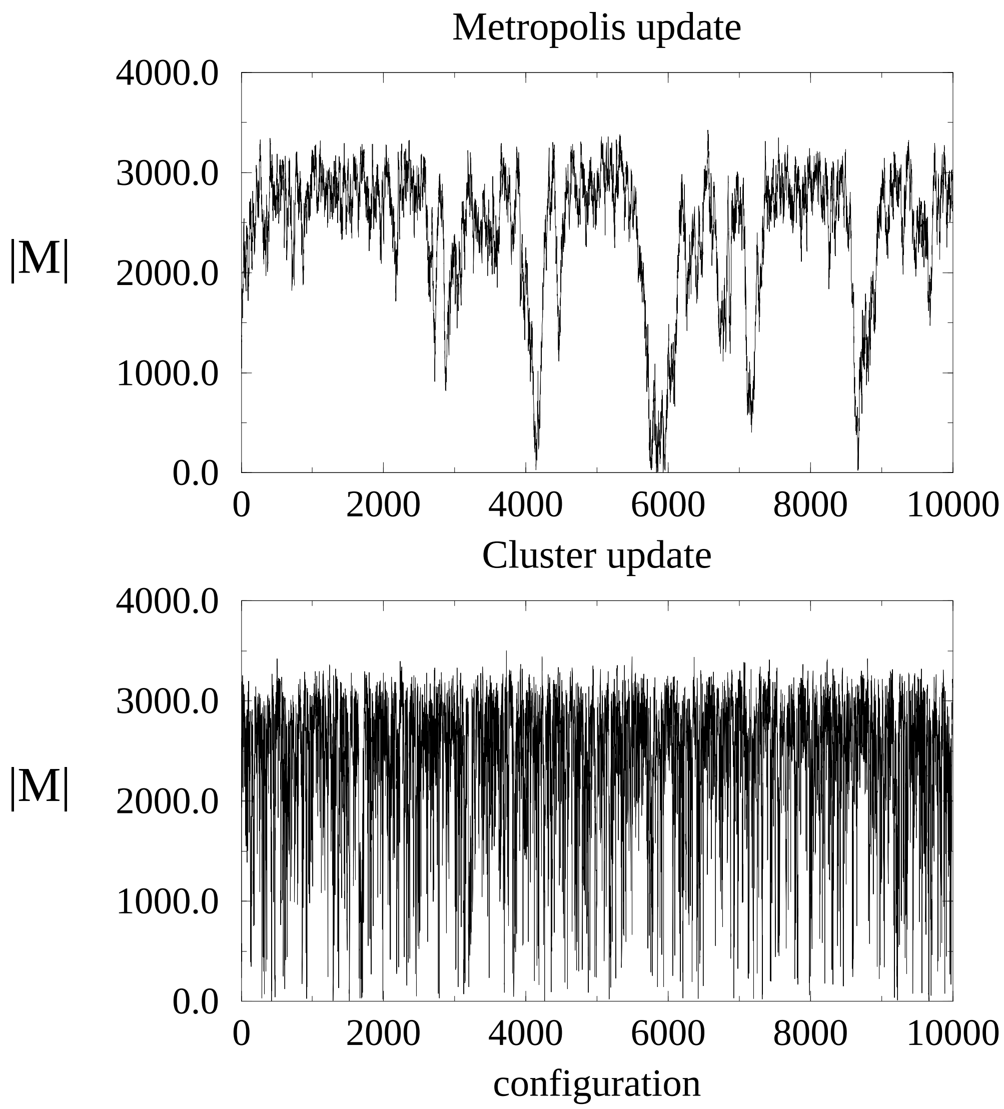
```
void update_cluster()
{
  int start,state;

  start = dran() * volume;    /* starting location */
  state = s[start];           /* starting spin value */

  grow_cluster(start, state); /* start growing and inverting the cluster */
}

void grow_cluster(int loc,int state)
{
  int i,new_loc;

  s[loc] = -s[loc];          /* invert the spin at this location */

  /* begin loop over neighbour locations */
  for (i=0; i<n_neighbours; i++) {
    new_loc = neighbour[i][loc];   /* array of neighbour indices */

    if (s[new_loc] == state && exp(-2.0*beta) < dran())
      grow_cluster(new_loc,state);
  }
}
```

• `s[loc]` is flipped during the cluster growth: this prevents revisiting sites already included in the cluster.

• Recursion (as above) is neat but not efficient! Better to unroll the recursion with loops and temporary arrays.

System size 64^2, $\beta = 0.44$. Compare measurements of absolute value of magnetization

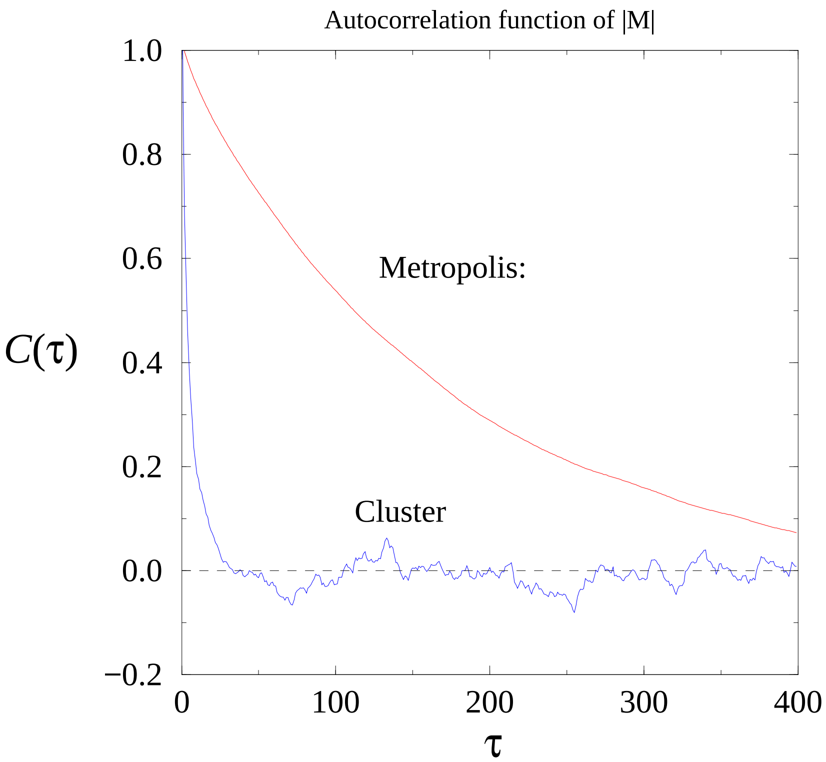
$$|M| = |\sum_i s_i|$$



Cluster update covers the phase space much faster than Metropolis!

The autocorrelation function describes how fast measurements become decorrelated.
Let now $O_i = |M|_i - < |M| >$, where $|M|_i$ is the measurement $\#i$ of $|M|$.
Autocorrelation function $C(\tau)$ of $|M|$:

$$C(\tau) \equiv \sum_i O_i O_{i+\tau} \quad \propto \quad \exp -\tau/\tau_a$$

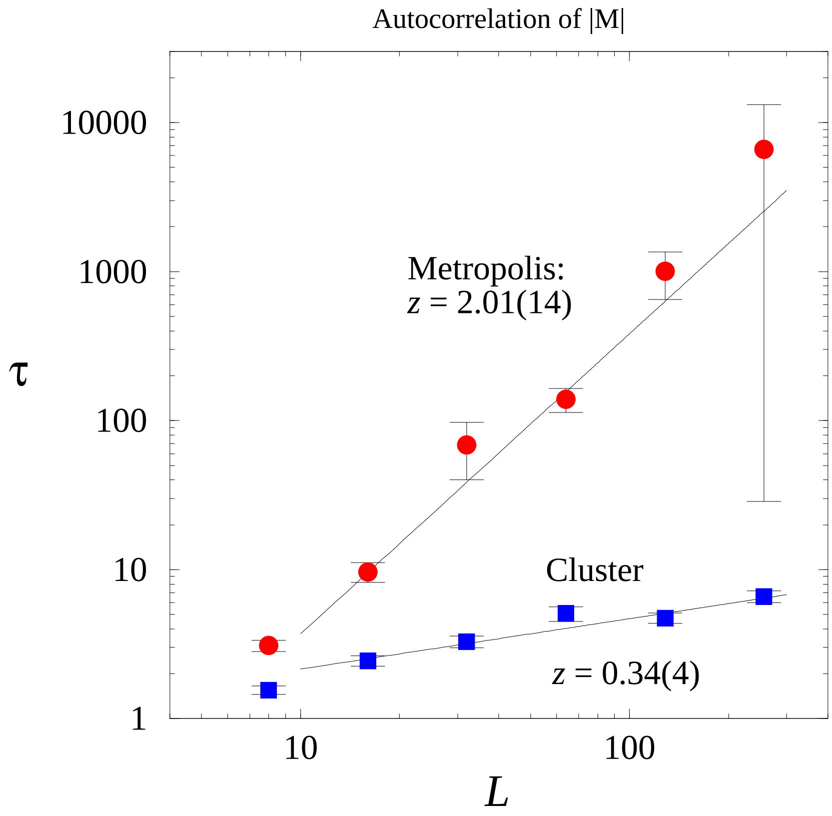


• Exponential autocorrelation time: the exponential decay length of $C(\tau)$ at large τ.

• Integrated autocorrelation time:

$$\tau_a^{\text{int.}} \equiv \frac{1}{C(0)} \int d\tau C(\tau) \, .$$

Autocorrelation time $\tau_a^{\text{int.}}$ at different lattice sizes $L^2 = 8^2 - 256^2$:



• Expected behaviour: $\tau_a \propto L^z$, where z is called dynamical critical exponent. For local (dissipative) algorithms, $z \geq \gamma/\nu \sim 2.125$ for the 2d Ising model.

• For volumes $\sim 256^2$, cluster algorithm is ~ 1000 times better than Metropolis!

• Measured in real (cpu/wallclock) time, cluster is even better (at the critical temperature $\beta_c = 0.44$):

time/update $\propto V$ (local Metropolis)

time/update $\propto V^x$, where $x < 1$

What about other models?

Cluster algorithms can work, if one can embed an Ising system into the original system. Example: O(N) non-linear sigma model:

The Hamiltonian of the model is

$$E = -\sum_{<ij>} \sigma_i \cdot \sigma_j \,, \qquad \text{where } |\sigma_i|^2 = \sum_{\alpha=1}^{N} (\sigma_i^\alpha)^2 = 1$$

Thus, spins σ_i are N-dimensional vectors of unit length.

Embedding: [U.Wolff, PRL 62 (1989) 361]

Map arbitrary sigma model configuration to Ising model with Hamiltonian

$$E_I = -\sum_{<ij>} J_{ij} s_i s_j$$

as follows:

- Choose a random O(N) vector r.
- Set all Ising model links $J_{ij} = (r \cdot \sigma_i)(r \cdot \sigma_j)$.
- Ising update $s_i \to -s_i$ corresponds to reflection of σ_i along the vector r: $\sigma_i \to \sigma_i - 2(r \cdot \sigma_i)\, r$
- Initially, all $s_i = +1$.

After the mapping, we can use either S-W or Wolff update on the Ising spins. In practice, one Wolff update cycle proceeds as follows:

1. Choose random O(N) vector r.
2. Choose random site i as the starting point for the cluster.
3. Study neighbouring sites j. Join site j to cluster with probability $p = 1 - \exp(-2\beta J_{ij})$, where $J_{ij} = (r \cdot \sigma_i)(r \cdot \sigma_j)$.
4. Repeat step 3 for all sites joined to the cluster. Keep on doing this as long as the cluster grows.

5. When the cluster is finished, reflect $\sigma_i \to \sigma_i - 2(r \cdot \sigma_i)$ for all sites which belong to the cluster.

Typically the reflection is performed during the cluster growth.

Single cluster O(N) sigma model algorithm works even better than for Ising: In 3-dimensional O(4) sigma model, the dynamical critical exponent z becomes negative.

In what models clusters work, in what models they fail?
Clusters (usually) fail, if

- there are frustrated couplings (spin glasses, gauge theories ...)
- one cannot construct a symmetric reflection operation
- spins are 'frozen' in place by external fields etc.

Cluster updates are (normally) usable only in proximity of a 2nd order phase transitions: large correlation lengths $\to$ large clusters.

Nevertheless, sometimes they are useful when correlation lengths are finite but still large ($\gg 1$) in lattice units.

In order to work, the reflection operation $R\sigma = \sigma'$ must satisfy the following: the set of the fixed points of the reflection ($R\sigma = \sigma$) must separate the phase space of σ in two disconnected sets.
For example, in O(3), the fixed points are the points with $r \cdot \sigma = 0$, i.e. the points on the 'equator' perpendicular to the reflection vector r. This divides O(3) into disconnected 'northern' and 'southern' hemisphere.
Mathematically: fixed point set should have codimension 1 [Lüscher, Weisz, Wolff, NPB 359 (1991) 221].

Reweighting

- MC simulation at $\beta = \beta_c$.
- Expectation value of O

$$\langle O \rangle_{\beta_c} \equiv \frac{1}{Z} \sum_{\{s\}} O(s)\, e^{-\beta_c E(s)} \approx \frac{1}{N} \sum_i O_i$$

- Reweighting allows us to estimate $\langle O \rangle_\beta$, if $|\beta - \beta_c|$ is sufficiently small.
- Basic principle: consider measurements O_i, E_i. Now

$$\frac{1}{N} \sum_i O_i e^{-(\beta-\beta_c)E_i} = \frac{\sum_{\{s\}} [O e^{-(\beta-\beta_c)E}]\, e^{-\beta_c E}}{Z_{\beta_c}}$$

$$\frac{1}{N} \sum_i e^{-(\beta-\beta_c)E_i} = \frac{\sum_{\{s\}} [e^{-(\beta-\beta_c)E}]\, e^{-\beta_c E}}{Z_{\beta_c}} = \frac{Z_\beta}{Z_{\beta_c}}$$

$$\mapsto \quad \boxed{\langle O \rangle_\beta = \frac{\sum_i O_i\, e^{-(\beta-\beta_c)E_i}}{\sum_i e^{-(\beta-\beta_c)E_i}}}$$

Note: $\sum_i$ goes over measurements, E_i, O_i must be measured from the same configuration.

In practice: during a MC run, write down all measurements E_i and various O_i in a file. After the run, use eq. $\square$ to calculate $\langle O \rangle_\beta$.

History:

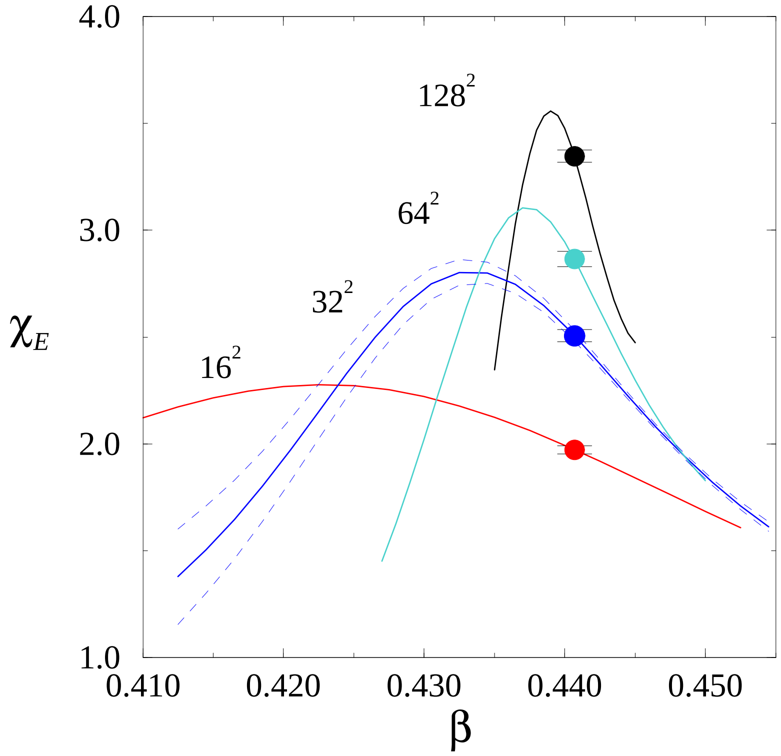
- First proposed in '59 [Salzburg et al, J.Chem.Phys 30 (1959) 60]
- First used by McDonald and Singer 1967, no success (reweighting range too large?)

• Shown to be very effective by Ferrenberg and Swendsen [PRL 61 (1988) 2635]
• Multihistogram method: F+S [PRL 63 (1989) 1195; Computers in Physics, Sep/Oct 1988]

Example: 2d Ising, $V = 16^2 \ldots 128^2$. Energy susceptibility (Heat capacity)

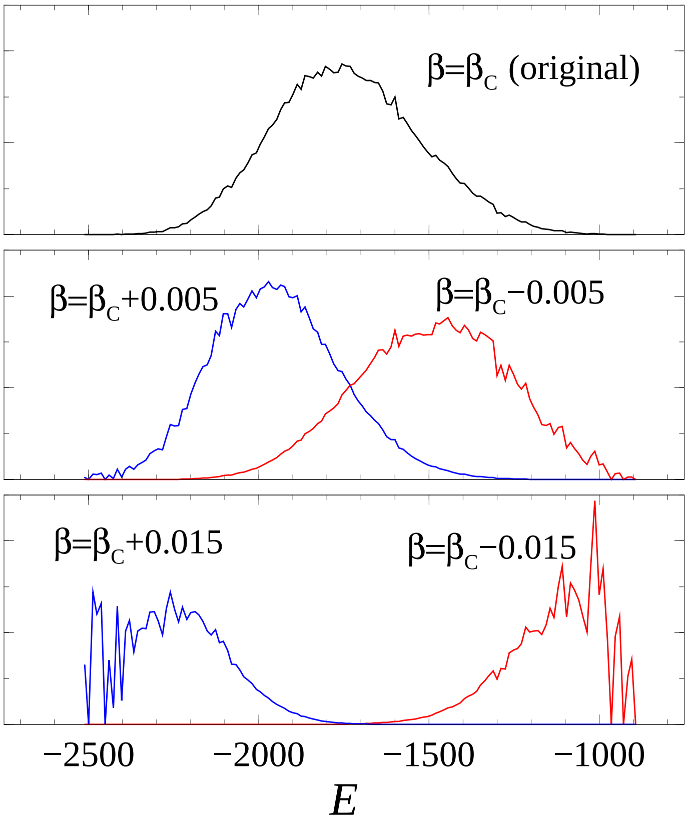
$$\chi_E = -\frac{1}{V}\frac{\partial\langle E\rangle}{\partial\beta} = \frac{1}{V}\langle(E - \langle E\rangle)^2\rangle$$

Susceptibility diverges with a critical exponent $\chi_E \sim L^{\alpha/\nu}$ when $L \to \infty$ and we are at $\beta = \beta_c$:



Points: simulated values, curves: reweighted data (with *jackknife* error estimates). Dashed lines show the error band for 32^2 (for clarity, not shown for others).

Energy histograms, 64^2 (probability distributions of E). Original simulation at $\beta = \beta_c \approx 0.44$.



The histograms become exponentially worse when the reweighting range increases.

How far one can reweight? Valid reweighting range $\ll$ 'fluctuation' range:

- Simulation at non-critical point: $\Delta\beta \propto 1/\sqrt{V}$.
- At a critical point (as in our example): $\Delta\beta \propto 1/L^x$, where x is some critical exponent. In our case above, $x = d - \alpha/\nu$ (I think), where $d = 2$ is the dimensionality.

Reweighting wrt. arbitrary parameters g^a, which couple to observables G^a: the partition function is

$$Z_g = \sum_s \exp[\sum_a g^a G^a]$$

Simulation is performed with $g^a = g^a_s$. Reweighting:

$$\langle O \rangle_g = \frac{\Sigma_i O_i \, \exp[\Sigma_a (g^a - g^a_s) G^a_i]}{\Sigma_i \exp[\Sigma_a (g^a - g^a_s) G^a_i]}$$

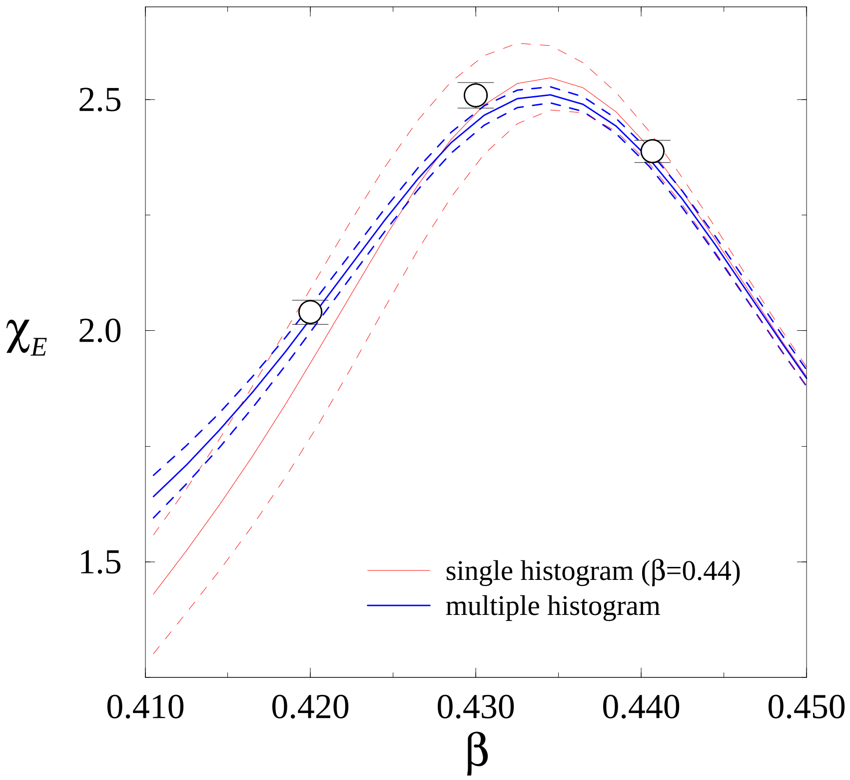
Thus, $g^1 = -\beta$, $G^1 = E$ recovers standard reweighting.

Finding Lee-Yang zeroes of the partition function: reweight to complex β, so that $Z_\beta = 0$.

Multiple histogram reweighting

Multiple histogram method joins together several sets of data, run at different β-values, in an optimized way.

Example:
Susceptibility χ_E, volume 32^2. The data from 3 runs is joined together with multiple histogram reweighting (blue curve). Errors are much smaller than with single histogram reweighting.



Powerful method – should be part of standard toolbox for everybody who can join points from different simulations. However, somewhat cumbersome to use.

- Perform R Monte Carlo simulations, at couplings β_i, with length N_i.
- Measure the energy distributions $p_i(E) = H_i(E)/N_i$, and the autocorrelation time τ_i.
- True distribution is given by

$$p_i(E) = n(E)\, e^{-\beta_i E + f_i},$$

where $n(E)$ is the density of states (does not depend on β), and f_i is (dimensionless) free energy: $f_i = -\log Z_{\beta_i}$.

$\mapsto$ Each of the simulations gives us an estimate of $n(E) = p_i(E)e^{\beta_i E - f_i}$ (we don't know f_i, so that the normalization is unknown). Since the MC runs were performed at different β_i, each of the run yields a reliable estimate of $n(E)$ only at limited range of E.

- **Optimization:** we obtain an improved estimate for $n(E)$ by combining all runs together:

$$n(E) = \sum_{i=1}^{R} r_i(E) p_i(E) e^{\beta_i E - f_i}\,, \quad \text{where} \quad \sum_{i=1}^{R} r_i(E) = 1 \text{ for all } E.$$

Note that the relative weights $r_i(E)$ are independent at each E: $\to$ optimization.

- $r_i(E)$ are determined by minimizing the $(\text{error})^2$ in $n(E)$ (now follows somewhat technical derivation):
  - What is the uncertainty in histogram values? Assuming that $H_i(E)$ is Poisson distributed around the 'true' value $\bar{H}_i(E)$, we obtain

$$\delta^2 H_i(E) = g_i \bar{H}_i(E) = g_i N_i n(E) e^{-\beta_i E - f}\,.$$

    Here $g_i = 1 + 2\tau_i$ takes into account the autocorrelations in run i.
  - Thus, $(\text{error})^2$ in $n(E)$

$$\delta^2 n(E) = \sum_i r_i^2(E) \frac{\delta^2 H_i(E)}{N_i^2} e^{2(\beta_i E - f_i)} = \sum_i r_i^2(E) \frac{g_i n(E)}{N_i} e^{\beta_i E - f_i}$$

– Minimize $\delta^2 n(E)$ wrt. $r_i(E)$ with condition $C \equiv \sum_i r_i(E) = 1$. Use Lagrange multipliers (try it):

$$\frac{\partial}{\partial r_i(E)}[\delta^2 n(E) + \lambda C] = 0 \quad \mapsto \quad r_i(E) = \frac{N_i g_i^{-1} e^{-\beta_i E + f_i}}{\sum_{j=1}^R N_j g_j^{-1} e^{-\beta_j E + f_j}}$$

- Thus, the optimized expression for $n(E)$ is

$$n(E) = \frac{\sum_{i=1}^R g_i^{-1} H_i(E)}{\sum_{j=1}^R N_j g_j^{-1} e^{-\beta_j E + f_j}}$$

- The coefficients f_i are then determined by solving

$$e^{-f_i} = \sum_E n(E) e^{-\beta_i E}$$

  To solve this equation use some iterative method (Newton-Raphson). f_i's are determined up to an additional constant.

- Observable expectation values:

$$\langle O \rangle_\beta = \frac{\sum_E O(E) n(E) e^{-\beta E}}{\sum_E n(E) e^{-\beta E}}$$

As with the single histogram method, this can be formulated without resorting to $H_i(E)$. Let E_i^a be the energy measurement number a ($a = 1 \ldots N_i$) from run number i. The expression for the free energy f_β becomes

$$e^{-f_\beta} = \sum_{i=1}^R \sum_{a=1}^{N_i} \frac{g_i^{-1} e^{-\beta E_i^a}}{\sum_{j=1}^R N_j g_j^{-1} e^{-\beta_j E_i^a + f_j}} .$$

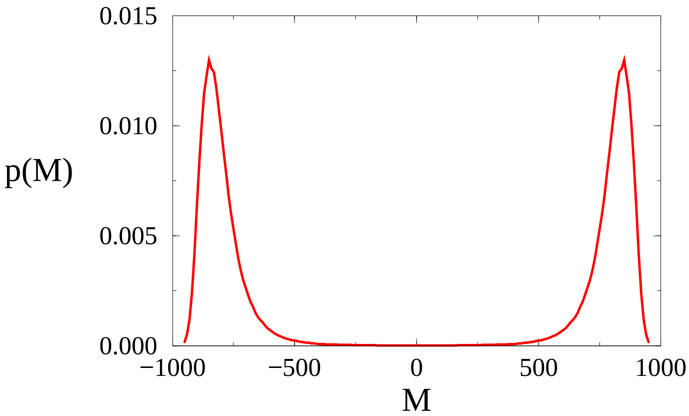
f_i's are then solved from $e^{-f_i} = e^{-f_{\beta_i}}$, in analogy to the second eq. in the box above. The expectation value of O at reweighted β:

$$\langle O \rangle_\beta = \sum_{i=1}^R \sum_{a=1}^{N_i} \frac{O_i^a g_i^{-1} e^{-\beta E_i^a - f_\beta}}{\sum_{j=1}^R N_j g_j^{-1} e^{-\beta_j E_i^a + f_j}}$$

Multicanonical Monte Carlo simulations

- Normal Monte Carlo algorithms sample configurations with the Boltzmann weight $p \propto \exp(-\beta E)$. Sometimes this is not desirable.
- Consider again Ising model, this time at $T < T_c$: now it has a 1st order phase transition between states with positive and negative magnetization.

  Magnetization distribution $p(M)$, volume 32^2, $\beta = 1/T = 0.453$, $h = 0$:



- Canonical probability

$$p_{\text{can}}(M) \propto \sum_{\{s\}} \delta(M, M_s) e^{-\beta E_s}$$

  is strongly suppressed between the peaks

  $\rightarrow$ canonical MC simulations hardly ever sample those configurations.

- Multicanonical method:

  Instead of using the Bolzmann weight $p \propto \exp[-\beta E]$ in Monte Carlo simulations, use a modified probability

$$p \propto \exp[-\beta E + W(M)]\,,$$

where $W(M)$ is a weight function, inserted 'by hand', which is carfully tuned to enhance the probability between the peaks.

- Multicanonical probability is produced by multicanonical simulations:

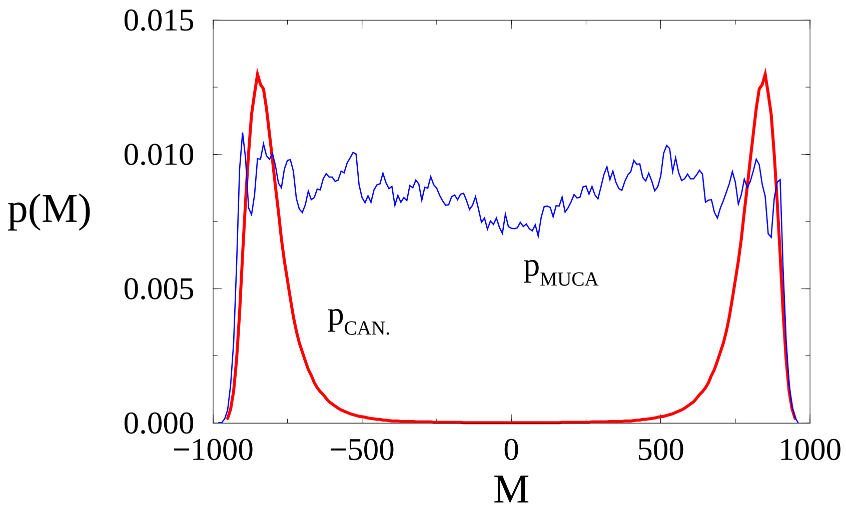
$$p_{\text{muca}}(M) \propto \sum_{\{s\}} \delta(M, M_s) e^{-\beta E_s + W(M_s)} \propto p_{\text{can}}(M) e^{W(M)}$$

- If we want that $p_{\text{muca}}(M)$ is flat in the region between the peaks we should choose $W(M) = -\log p_{\text{can}}(M)$. However, we do not a priori know $p_{\text{can}}(M)$. Thus, one has to do with approximations (later).

- Multicanonical simulation gives us $p_{\text{muca}}(M)$. We obtain the canonical distribution (which is, after all, what we're after!) by reweighting:

$$p_{\text{can}}(M) \propto p_{\text{muca}}(M) e^{-W(M)}$$

- The picture below shows $p_{\text{muca}}(M)$, which is $\sim$ constant. The wigglines is due to numerical noise (statistics) and the 'incorrectness' of $W(M)$. p_{can} has been obtained from p_{muca} by reweighting.



- How accurately should W be determined? Remember that the simulation is formally correct with arbitrary $W(M)$. Good W can just save a lot of cpu-time. We should only demand that $p_{\text{muca}}(M)$ does not become too small in any area of interest.

  $\mapsto$ If we allow p_{muca} to vary $\sim 50\%$, W can vary by $W \pm \log 2$.

- The parametrization of $W(M)$ is not unique. Perhaps the most common is a piecewise linear one: choose a discrete small set $\{m_i\}$ of magnetizations, and parametrize $w_i = W(m_i)$, and interpolate linearly between m_i's:

$$W(M) = w_{i+1}\frac{M - m_i}{m_{i+1} - m_i} + w_i\frac{m_{i+1} - M}{m_{i+1} - m_i} \qquad \text{when } m_i \le M \le m_{i+1}\,.$$

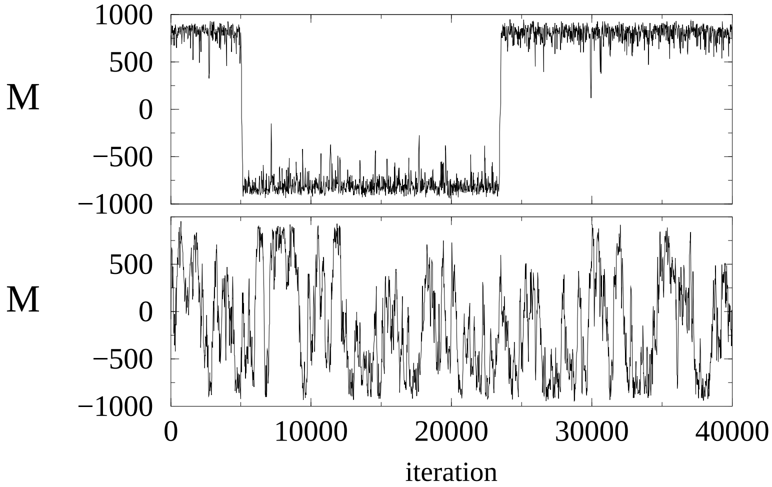
  There are several (semi)automatic iterative methods in the market for determining the weight function W.

- In the example above W was a function of M ("multimagnetic" ensemble). In general, W can be a function of any order parameter. Original formulation was in terms of E (useful for temperature-driven transitions).

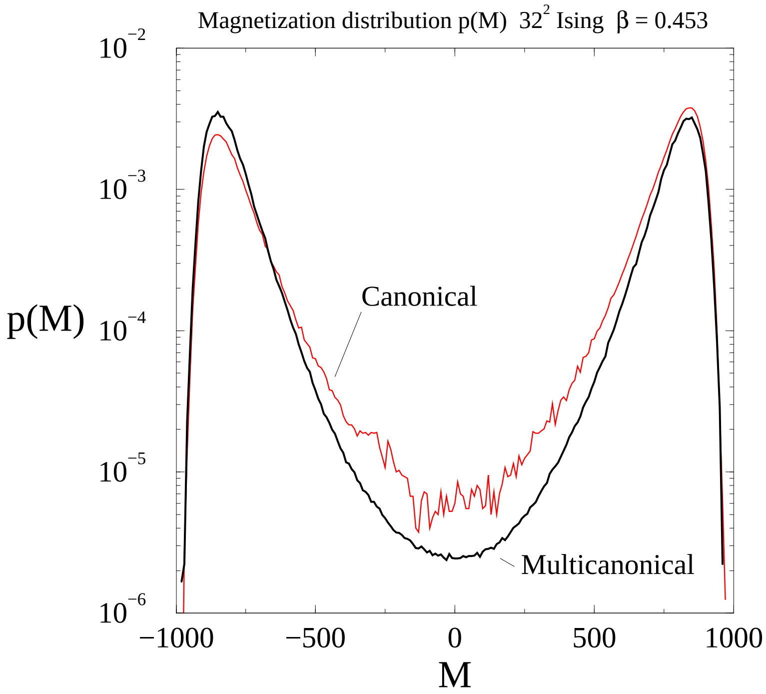
- Multicanonical [Berg, Neuhaus, Phys.Lett.B 267 (1991)]
  Umbrella sampling [Torrie, Valleau, J.Comp.Phys. 23 (1977)]

Performance comparison: again 32^2 Ising, at $\beta = 0.453$.

- Monte Carlo time history of M, from canonical and multicanonical simulations:



- Canonical distributions $p_{\mathrm{can}}(M)$, obtained from multicanonical and canonical simulations (200000 measurements):



• In 1st order phase transitions, the probability suppression between the pure phase peaks increases exponentially. The configurations in the middle are 'mixed': they contain a fraction of both pure phases, with an interface in between. The interface has extra free energy, $f_A = \sigma A$, where σ = surface tension, and A is the area of the interface. Minimally $A = 2L$ on 2-dim. lattice with periodic boundaries.

$$p_{\min}/p_{\max} \sim e^{-\sigma 2L}$$

Example:
20-state 2-d Potts model:

$$E = \sum_{ij}(1 - \delta_{s_i,s_j})$$
$$s_i = 1 \ldots 20$$

Temperature-driven 1st order phase transition at $\beta_c = \log(1 + \sqrt{20}) \approx 2.900$

[A. Billoire, T. Neuhaus and B. Berg, 1994]

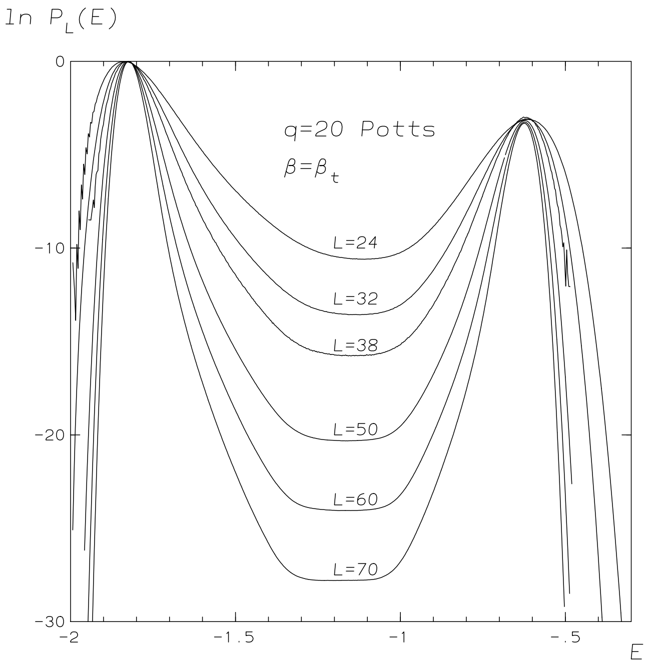
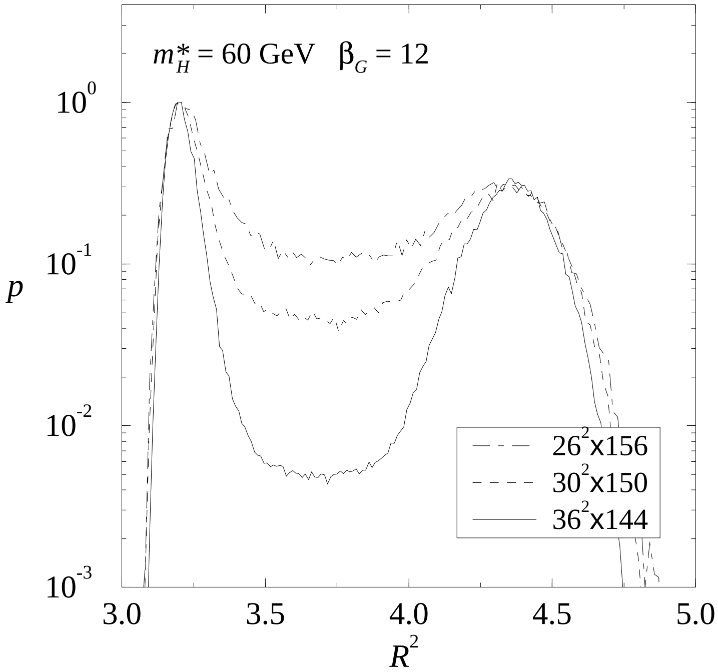


FIG. 8

Multicanonical simulations are not sensitive to this suppression. However, they have to 'random walk' through the transition region. The region widens $\propto V$, but the 'step size' (change in one update cycle) increases only as $\propto V^{1/2}$:

- $\tau \sim e^{-\sigma 2L}$ in canonical simulations (supercritical slowing down)
- $\tau \sim V^2 = L^{2d}$ in multicanonical simulations

SU(2)-Higgs model (Standard Model phase transition at $T \sim 100$ GeV.



[Kajantie, Laine, K.R, Shaposhnikov, 1995]

Strong 1st order transition if Higgs particle mass $\lesssim 60$ GeV.