A Modified Heat Bath Method Suitable for Monte Carlo
Simulations on Vector and Parallel Processors

K.Fredenhagen¹

II. Institut für theoretische Physik, Luruper Chaussee 149,
2000 Hamburg 50, West Germany

M.Marcu

II. Institut für theoretische Physik, Hamburg
and
CERN, CH-1211 Geneva 23, Switzerland

Abstract: We describe a modification of the heat bath algorithm for Monte Carlo simulations. In the new algorithm the same operations are performed for each update also in the case when an a priori probability has to be generated by an acceptance-rejection method. Thus the algorithm can always be easily implemented on vector or parallel processors.

¹ Heisenberg fellow

Monte-Carlo methods are widely used to simulate systems of interest to Solid State and Elementary Particle Physics[1]. The aim is to compute expectation values of functions on some huge probability space (Ω, μ) :

$$\langle f \rangle = \int f(\sigma) d\mu(\sigma)$$

Ω is typically the set of configurations σ of some fields defined on a lattice and μ is the Boltzmann distribution with respect to the energy functional $S(\sigma)$,

$$d\mu(\sigma) = \rho(\sigma) d\sigma, \quad \rho(\sigma) = \exp(-S(\sigma)) / \int \exp(-S(\sigma)) d\sigma \quad (2)$$

Even on small lattices the integrals cannot be evaluated exactly. The Monte Carlo method uses an ergodic Markov process [2] to estimate (1) numerically. Let P be a stochastic matrix on (Ω, μ) :

$$P(\sigma', \sigma) > 0, \quad \int P(\sigma', \sigma) d\sigma = 1, \quad \int P(\sigma', \sigma) \rho(\sigma) d\sigma = \rho(\sigma') \quad (3)$$

For a sequence of configurations $\{\sigma_i, i=0, \dots, N\}$ generated using $P(\sigma_{i+1}, \sigma_i)$ as a transition probability density (i.e. for a Markov chain) we have with probability one:

$$\langle f \rangle = \lim_{N \rightarrow \infty} \frac{1}{N - N_0} \sum_{i=N_0+1}^N f(\sigma_i) \quad (4)$$

In practice one generates a finite Markov chain using a computer and chooses σ_0 , N_0 and N such that the statistical errors are as small as desired.

A configuration σ can be seen as a set of real variables x_α that are well localized on the lattice (e.g. on sites and/or links). Denoting $\tilde{x}_\alpha = \sigma \setminus \{x_\alpha\}$, $p(\sigma)$ can be written as

$$p(\sigma) d\sigma = p_{\tilde{x}_\alpha}(x_\alpha) dx_\alpha \tilde{p}(\tilde{x}_\alpha) d\tilde{x}_\alpha \quad (5)$$

$p_{\tilde{x}_\alpha}(x_\alpha)$ is the conditional transition probability density for x_α with $\tilde{x}_\alpha$ fixed. In order to devise an efficient algorithm P is defined as either the product of local stochastic matrices (ordered sweep through the lattice),

$$P = \prod_{\alpha} P_{\alpha}, \quad P_{\alpha}(\sigma', \sigma) = P_{\tilde{x}_\alpha}(x'_\alpha, x_\alpha) \delta(\tilde{x}'_\alpha - \tilde{x}_\alpha) \quad (6)$$

or a weighted sum of local stochastic matrices (random choice of the variable to be updated),

$$P = \sum_{\alpha} w_{\alpha} P_{\alpha}, \quad \sum_{\alpha} w_{\alpha} = 1 \quad (6')$$

(P_{α} are the same as in (6)). $P_{\tilde{x}_\alpha}(x'_\alpha, x_\alpha)$ is the conditional probability density from σ to σ' with $\tilde{x}_\alpha$ fixed. A configuration is updated by implementing a sequence of one-variable ergodic Markov processes. From now on we will drop the index α when discussing such a one-variable process.

The heat bath method introduced in [3] (see also [4]) consists in choosing

$$P_{\tilde{x}}(x', x) = p_{\tilde{x}}(x') \quad (7)$$

It is well established that for continuous variables this is more efficient than the more widely used Metropolis method [1,3,5-7]. Assume now that both the integral η_x of ρ_x and its inverse η_x^{-1} are known explicitly inside the interval (a,b) where x is defined. Then

$$\rho_x(x) dx = d\eta_x(x) = d\{\eta_x(a) + r(\eta_x(b) - \eta_x(a))\} \quad (8)$$

with $r \in (0,1)$. Taking for r uniformly distributed pseudo-random numbers,

$$x(r) = \eta_x^{-1}(\eta_x(a) + r(\eta_x(b) - \eta_x(a))) \quad (9)$$

simulates the desired distribution.

In many cases it is impossible to find or impractical to use η_x and η_x^{-1} explicitly. One may then apply von Neumann's acceptance-rejection method [3,4]. There one starts the simulation of ρ_x by simulating an approximate distribution $\rho_x^{(0)}$ for which $x(r)$ can be easily computed using (9) (see [3,5-8] for examples). $x(r)$ is accepted as the updated value of the stochastic variable x if for a second uniform pseudo-random number $r' \in (0,1)$

$$r' < \lambda_x \rho_x(x(r)) / \rho_x^{(0)}(x(r)) \quad (10)$$

where λ_x is chosen such that

$$\lambda_x \leq \rho_x^{(0)}(x) / \rho_x(x) \quad (11)$$

for all $x \in (a,b)$. Otherwise x maintains its former value.

The stochastic matrix for this process is no longer $P_{\tilde{x}}$ but

$$P_{\tilde{x}}^{(1)} = \lambda_{\tilde{x}} P_{\tilde{x}} + (1 - \lambda_{\tilde{x}}) \mathbb{1} \quad (12)$$

$\lambda_{\tilde{x}}$ is the acceptance rate for a change of x . The acceptance-rejection method now consists in repeating the update attempt described above until a change is accepted. The stochastic matrix $P_{\tilde{x}}^{(n)}$ for n attempts satisfies the equation

$$P_{\tilde{x}}^{(n)} = \lambda_{\tilde{x}} P_{\tilde{x}} + (1 - \lambda_{\tilde{x}}) P_{\tilde{x}}^{(n-1)}, \quad P_{\tilde{x}}^{(0)} = \mathbb{1} \quad (13)$$

which has the unique solution

$$P_{\tilde{x}}^{(n)} = \lambda_{\tilde{x}}^{(n)} P_{\tilde{x}} + (1 - \lambda_{\tilde{x}}^{(n)}) \mathbb{1}, \quad \lambda_{\tilde{x}}^{(n)} = 1 - (1 - \lambda_{\tilde{x}})^n \quad (14)$$

Repeating the update attempts until a change is accepted means taking the limit $n \rightarrow \infty$:

$$P_{\tilde{x}}^{(\infty)} = \lim_{n \rightarrow \infty} P_{\tilde{x}}^{(n)} = P_{\tilde{x}} \quad (15)$$

This is the stochastic matrix of the heat bath method [3,4].

If instead of using the heat bath method, one stops after n attempts, the process is described by the stochastic matrix $P_{\tilde{x}}^{(n)}$ which has the same stationary distribution $\rho_{\tilde{x}}$ as $P_{\tilde{x}}$

$$\int P_{\tilde{x}}^{(n)}(x', x) \rho_{\tilde{x}}(x) dx = \rho_{\tilde{x}}(x') \quad (16)$$

It even satisfies detailed balance

$$P_{\tilde{x}}^{(n)}(x', x) \rho_{\tilde{x}}(x) = P_{\tilde{x}}^{(n)}(x, x') \rho_{\tilde{x}}(x') \quad (17)$$

if $P_{\tilde{x}}$ also satisfies this sufficient condition for (16). Thus terminating the update attempts after n steps is also an admissible Monte Carlo procedure.

The same Markov process can be simulated by iterating n times the process described by $P_{\tilde{x}}^{(1)}$. This means that an attempt to update x using a pair of random numbers r, r' is done irrespective of whether a change of x was accepted in an earlier attempt. Actually in this case the stochastic matrix is $(P_{\tilde{x}}^{(1)})^n$, but for the heat bath process (7) $P_{\tilde{x}}$ is a projector

$$P_{\tilde{x}}^2 = P_{\tilde{x}} \quad (18)$$

which implies

$$(P_{\tilde{x}}^{(1)})^n = P_{\tilde{x}}^{(n)} \quad (19)$$

For the whole system one obtains (for both procedures) as stochastic matrix for an n -attempt update

$$P^{(n)} = \prod_{\alpha} P_{\alpha}^{(n)}, \quad P_{\alpha}^{(n)}(\sigma', \sigma) = P_{\tilde{x}_{\alpha}}^{(n)}(x'_{\alpha}, x_{\alpha}) \delta(\tilde{x}'_{\alpha} - \tilde{x}_{\alpha}) \quad (20)$$

or

$$P^{(n)} = \sum_{\alpha} w_{\alpha} P_{\alpha}^{(n)} \quad (20')$$

$p^{(n)}$ satisfies (3). This procedure is easy to implement on vector and parallel processors since the same operations are performed for the update of each variable.

In order to implement Creutz's original method [3] on a vector processor, the update of a configuration is split into a vectorized part and a scalar part as explained in [5]. There the vectorized updates are done until the acceptance rates $\lambda_{\tilde{x}_\alpha}$ are close to one for all α (technically, a number of sweeps through appropriately chosen sublattices is performed). Then the update attempts are continued in a non vectorized way for those variables which still have the old values, until every one of them is changed. Our above considerations show that this scalar part is unnecessary. Furthermore, there is no need to take n so large that $\lambda_{\tilde{x}_\alpha}$ are very close to one. Actually as an empirical fact these two points were already noticed in [6].

An optimal value n_{opt} of n can be found by considering the average acceptance rate

$$a^{(n)} = \langle \lambda_{\tilde{x}_\beta} \rangle = \langle \sum_{\alpha} \lambda_{\tilde{x}_\alpha} \rangle / M \quad (21)$$

where the sum on the r.h.s. is done over the M values of α for which x_α are updated in the same way as x_β . Typically $a^{(n)}$ can be determined in short trial simulations. Two bounds can be helpful in determining n_{opt} :

$$1 - (1 - \lambda_{\min})^n \leq a^{(n)} \leq 1 - (1 - a^{(1)})^n. \quad (22)$$

$\lambda_{\min}$ is chosen such that $\lambda_{\min} \leq \lambda_{\tilde{x}_\alpha}$ for all α summed over in (21). In practice it is found by assuming a worst-case scenario for the variables neighbouring x_α . In general we expect that the statistical errors will decrease with increasing values of n . However this gain will be upset by the loss in computer time if n exceeds n_{opt} . It is by no means true that $a^{(n_{\text{opt}})}$ has to be very close to one. In fact in the case of the SU(2) Higgs model (see [9] for a review) we found $a^{(n_{\text{opt}})}$ to be of the order .7-.9 (in the Higgs region of the model) when using a vectorized program on a CRAY 1/M computer. During the same simulation we observed that the second inequality in (22) was almost an equality (up to the third digit).

The heat bath method can always be applied unless the parametrization of a configuration with real variables is too cumbersome. The distribution $p_{\tilde{x}}$ can be approximated by piecewise linear functions [8] or by other simple functions like piecewise polynomial and/or power and/or exponential. By choosing n finite the algorithm can be easily vectorized. A lot of flexibility is gained by not trying to push the acceptance rate very close to one. Thus for example, worse but faster and easier to program approximations of $p_{\tilde{x}}$ may be chosen.

!

Acknowledgements

We would like to thank T.Filk, F.Gutbrod, I.Montvay and I.O.Stamatescu for helpful discussions.

References

- [1] M.Creutz, L.Jacobs and C.Rebbi, Phys.Repts.95 (1983) 201
K.Binder, J.Comp.Phys.59 (1985) 1
- [2] M.S.Bartlett, "An Introduction to Stochastic Processes", Cambridge University Press (1978)
- [3] M.Creutz, Phys.Rev.D21 (1980) 2308
- [4] R.Y.Rubinstein, "Simulation and the Monte Carlo Method", Wiley(1981)
- [5] P.de Forcrand, D.Lellouch and C.Roiesnel, J.Comp.Phys.59 (1985) 324
A.Kennedy et.al., J.Comp.Phys.64 (1986) 133
- [6] K.C.Bowler and B.J.Pendleton, Nucl.Phys.B220 (1984) 109
- [7] K.Fabricsius and O.Haan, Phys.Lett.143 (1984) 459
- [8] F.Gutbrod, unpublished
S.Miyashita, Prog.Theor.Phys.63 (1980) 797
- [9] J.Jersak in "Lattice Gauge theory - a Challenge to Large Scale Computing", B.Bunk, K.H.Mutter and K.Schilling eds., Plenum (1986)