Stochastic cluster algorithms for discrete gaussian (SOS) models

Hans Gerd Evertz [a,b], Martin Hasenbusch [c], Mihail Marcu [b,d], Klaus Pinn [b,e] and Sorin Solomon [f]

[a] *Supercomputer Computations Research Institute, Florida State University, Tallahassee, FL 32306, USA*
[b] *II. Institut für Theoretische Physik, Universität Hamburg, Luruper Chaussee 149, W-2000 Hamburg 50, FRG*
[c] *Fachbereich Physik, Abteilung Theoretische Physik, Universität Kaiserslautern, Postfach 3049, W-6750 Kaiserslautern, FRG*
[d] *School of Physics and Astronomy, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, 69978 Tel Aviv, Israel*
[e] *Institut für Theoretische Physik I, Universität Münster, Wilhelm-Klemm-Straße 9, W-4400 Münster, FRG*
[f] *Racah Institute of Physics, Hebrew University, 91904 Jerusalem, Israel*

Received 19 October 1990

We present new Monte Carlo cluster algorithms which eliminate critical slowing down in the simulation of solid-on-solid models. In this letter we focus on the two-dimensional discrete gaussian model. The algorithms are based on reflecting the integer valued spin variables with respect to appropriately chosen reflection planes. The proper choice of the reflection plane turns out to be crucial in order to obtain a small dynamical exponent z. Actually, the successful versions of our algorithm are a mixture of two different procedures for choosing the reflection plane, one of them ergodic but slow, the other one non-ergodic and also slow when combined with a Metropolis algorithm.

1. Introduction

Computer simulations have become a major tool for investigating statistical mechanical systems and lattice field theories. The main limiting factor in such studies is critical slowing down (CSD). To overcome this problem, various proposals for nonlocal Monte Carlo algorithms were put forward over the years, one of the most successful approaches being that of cluster algorithms [1–4].

In this letter we describe a class of cluster algorithms for the discrete gaussian model ($\mathbb{Z}$-ferromagnet) defined by the partition function

$$\mathscr{Z}=\sum_{m}\exp\left(-\frac{1}{2\beta}\sum_{\langle x,y\rangle}(m_x-m_y)^2\right), \qquad (1)$$

where x and y are sites of a square lattice (with periodic boundary conditions), m is an integer valued field (spin) which takes the value $m_x \in \mathbb{Z}$ at the site x, and the sum in the exponential runs over all unordered nearest neighbor pairs $\langle x, y\rangle$. This model is also called the SOS (solid-on-solid) model, and it is a realistic model of a crystal surface (m_x describes the surface height at the two-dimensional point x) [5].

The model (1) is the dual of the two-dimensional XY model with Villain (heat kernel) actıon. For $\beta<\beta_c$ the global $\mathbb{Z}$-symmetry is spontaneously broken. As a consequence, there is a nonzero mass gap, and the (SOS) surface thickness, defined as the square root of $\lim_{|x-y|\to\infty}\langle (m_x-m_y)^2\rangle$, is finite. According to the Kosterlitz–Thouless (KT) scenario [6], the correlation length diverges exponentially as β approaches β_c. In the SOS terminology this is due to the fact that the surface fluctuates more and more. At the critical point the surface becomes rough; for all $\beta>\beta_c$ the surface thickness diverges logarithmically with the volume of the system. The large-β phase corresponds to the spin wave (massless) phase of the XY model.

Although there have been numerous numerical investigations of the KT transition, there is still no completely conclusive confirmation of the KT scenario. In order to supplement existing studies (see e.g. ref. [2], and, for a recent finite-size scaling analysis and further references, see ref. [7]), a fast algorithm for large scale simulations in the SOS representation is highly desirable. Furthermore, SOS models are interesting in their own right, and the algorithms we describe here can be applied to related surface models.

This letter is organized as follows. In section 2 we describe the general ideas and formal aspects of our cluster algorithms. Section 3 deals with the proper choice of the reflection plane, which is the crucial ingredient in speeding up the algorithm. Simultaneously, the main results concerning the autocorrelation times are discussed. In section 4 we attempt to give a physical explanation of why our algorithms are successful in overcoming CSD. Conclusions are given in section 5.

2. The cluster algorithms

The picture of the critical physics in the discrete gaussian model can be obtained by visualizing a configuration as a relief landscape over a plane base represented by the lattice. The integer field variable m_x is the height of the landscape at the point x. The objects dominating the critical dynamics are hills and valleys. The usual Metropolis algorithm, offering at each step a change of a single variable, is of course very inefficient in moving the hills and valleys around.

One way to perform large scale changes of a configuration is to select an arbitrary point on the landscape, draw a horizontal plane through it, consider the connected regions above and below the plane, and reflect them (with respect to the plane) independently, with an appropriate probability. In order to implement this qualitative picture in a rigorous way, consistent with the Boltzmann distribution, we propose the following cluster algorithm.

Let us denote the height of the horizontal *reflection plane* by M. A reflection of m_x with respect to M means

$$m_x \to 2M - m_x \,. \tag{2}$$

Obviously, M has to be either an integer or a half-integer. One way of explaining how the clusters are built is in terms of the *embedded Ising variables* [1] $\sigma_x = \pm 1$, defined by the decomposition

$$m_x = \sigma_x |m_x - M| + M \,. \tag{3}$$

$\sigma_x = 1$ means that m_x is above the reflection plane M, $\sigma_x = -1$ that it is below M. Note that σ_x is not well defined if $m_x = M$; this is not going to cause difficulties, as can be seen from eq. (4) below.

In order to define the cluster procedure we introduce the *deleting* and *freezing* probabilities [3] for a link $\langle x, y\rangle$ (in the language of ref. [1] the term "activating" is used instead of "freezing"):

$$P_{\text{del}}(x,y) = q \times \exp\left(-\frac{1}{\beta} |m_x - M|\,|m_y - M|\,(\sigma_x\sigma_y + 1)\right), \tag{4}$$

where $q \leqslant 1$ can explicitly depend on $|m_x - M|$ and $|m_y - M|$, and

$$P_{\text{freeze}}(x,y) = 1 - P_{\text{del}}(x,y) \,. \tag{5}$$

In contrast to other cluster algorithms investigated in the literature, the possibility of choosing $q \neq 1$ will prove to be useful in our case. Let us however assume for the moment that $q = 1$.

After freezing or deleting all the links of the lattice with the above probabilities, two sites are defined to be connected if they are at the endpoints of a frozen link. The clusters are then defined to be the connected components of the lattice.

Notice that $P_{\text{del}}(x,y) = 1$ if either $\sigma_x \neq \sigma_y$, i.e. m_x and m_y are on different sides of the reflection plane, or if $|m_x - M|\,|m_y - M| = 0$, i.e. at least one of the points lies on the reflection plane itself. Thus, similarly to the Ising model, the clusters will contain only spins for which the embedded Ising variables have the same value. On the other hand, the spins with $m_x = M$ are always monomers. The most important difference to the Ising model is however the strong dependence of the clusters on the choice of M. Consider for example a situation where M lies above most of the m_x. Since $P_{\text{del}}(x, y)$ becomes exponentially small with increasing distance from M, there will be with high probability one very large cluster, containing almost all spins.

Flipping a cluster means flipping the embedded Ising variables. In terms of the original integer variables m_x, this is equivalent to performing the reflection (2) for all spins in the cluster. Notice that the cluster boundaries are not in general exactly at the intersection of the relief landscape with the reflection plane, since $P_{\text{del}}(x, y)$ is nonzero also if both m_x and

m_y are away from and on the same side of M. Nevertheless, the intuitive picture of clusters as hills or valleys which are flipped through the reflection plane is approximately realized.

In our simulations we used the single cluster algorithm [2], i.e. a cluster is built starting from a randomly chosen site (the *seed*), and it is flipped with probability one.

In order to establish a valid algorithm one has to ensure detailed balance. Once M is given, detailed balance follows from standard arguments [3,2] for the restricted set of configurations related by reflecting the clusters with respect to M. A sufficient condition for detailed balance to hold for the entire procedure is to choose M with an a priori probability $\mathrm{prob}(M)$ that is a function of M itself and of the objects that are unchanged by the reflection, i.e. of the values of $|m_x - M|$ for all lattice sites:

$$\mathrm{prob}(M) = f(|m - M|; M) . \qquad (6)$$

This condition still leaves a lot of freedom in the choice of M. Note that if one starts with an unnormalized distribution of the form (6), one has to make sure that the normalization constant does not depend on m.

It turned out that the proper choice of the reflection plane is the crucial ingredient in overcoming CSD. We shall proceed now to the discussion of our choices for M and of our main simulation results.

3. Choice of the reflection plane and results

Using the picture of a configuration as a relief landscape, with hills and valleys, we were at the beginning led to the following choice for M. First we select randomly a lattice site x_0 as the seed of the cluster to be built. Typically x_0 will be neither on the top of a hill, nor on the bottom of a valley, but on a slope. If the reflection plane M is chosen close to m_{x0}, there is a good chance that the cluster algorithm will cut a hilltop or a valley of reasonable size from the landscape and flip it. The simplest choice of M leading to an ergodic algorithm is, in this framework, to take $M = m_{x0} \pm \frac{1}{2}$, each of the two possibilities with probability $\frac{1}{2}$, and $q=1$ (see eq. (4)). Taking $M = m_{x0}$ is not helpful for $q=1$ since all the links starting from x_0 would be deleted with probability one. Let us denote the procedure just described as the *H-algorithm* (H for half-integer).

We tested the H-algorithm extensively. In order to make sure that we do observe the slowest modes of the Markov matrix [4], we measured a whole range of quantities: mean energy and specific heat, surface thickness, various block spin correlation functions, and the order parameters $\mathcal{M}_\alpha \equiv \langle V^{-1} | \Sigma_x \exp(2\pi \mathrm{i} \times \alpha m_x) | \rangle$ for $\alpha = 0.1, 0.2, ..., 0.5$ (V is the volume of the lattice; $\mathcal{M}_\alpha$ is a good finite size approximation for the square root of $\lim_{|x-y| \to \infty} \langle \exp[2\pi \mathrm{i} \alpha (m_x - m_y)] \rangle$).

For all these quantities we studied the autocorrelation function and tried to determine the *exponential autocorrelation time* τ. Not that τ should in principle be the same for all our quantities, since it only depends on the second largest eigenvalue of the Markov matrix [4]. In practice, there always were quantities that did not couple well to the slow modes and exhibited a misleadingly small τ. Sometimes some quantities showed a clear crossover from one clean exponential regime to a slower second one. In other instances there were quantities that did not decay at all exponentially until the limits of our precision were reached. These difficulties notwithstanding, we were usually able to reach a clearcut conclusion with respect to the "true" value of τ because there was a whole set of "slowest decaying quantities" which exhibited this value.

The autocorrelation time τ is always quoted in "works units" (sweeps). A work unit is the work necessary to build a cluster of the size of the entire lattice. Each of our runs consisted of between 100 000 and 500 000 work units.

Determining the errors on τ is a delicate business. First we plotted the autocorrelation functions with error bars and decided, for each quantity, at what value of the Monte Carlo time a clean exponential regime started. In practice, we chose a logarithmic scale for the y-axis and looked for the onset of a linear regime. Then we drew the highest and the lowest slope straight lines compatible with the data that were deemed to be in that regime. The value quoted for τ is the average of the inverses of the two slopes. The errors are quoted such that the inverses of the highest and the lowest slope are at the ends of the interval value ± error.

The canonical procedure for determining τ and its error is different. After deciding by hand the extent of the clean exponential regime, we have to actually fit the data with an exponential. For this purpose, a least-square-fit procedure can be used. It is however crucial to take into account the covariances between the values of the autocorrelation function at different times (these covariances are by no means small). Otherwise it is not only that the estimated error bars for τ are unreliable, but we also lack a meaningful goodness-of-fit test. Then, we also have to check for consistency with respect to varying the time value for which the linear regime sets on. Such an analysis is in progress and we checked that the results for τ are consistent with those obtained using the highest–lowest slope method. As expected, the values of the errors are consistently smaller when using the least-square-fit method. The results of a more complete analysis, using both methods, will be published elsewhere.

We performed simulations in two physically distinct regimes:

- At $\beta=1$, deep in the massless (rough) phase. Since the correlation length ξ is infinite, the only scale is set by the linear extent L of the lattice. The dynamical critical exponent z is defined by $\tau \sim L^z$ [8].
- In the massive (smooth) phase at large ξ, i.e. close to the KT transition. In this regime, z is defined by the finite-size scaling law $\tau = \xi^z F(\xi/L)$ [7], where F is some unknown universal function which we could in principle measure.

In the smooth phase we estimated ξ by analyzing the exponential decay of the two-point function $\langle (m_x - m_y)^2 \rangle$ (appropriate subtractions, Fourier transforms, etc., were done in standard fashion). In this phase the particle states are *kinks* [#1], which correspond to one-dimensional (time-zero) configurations that have an integer value n at minus infinity and $n+1$ at plus infinity. With periodic boundary conditions however, there are in the Hilbert space of the problem only states containing kink–antikink pairs. Thus we tried to determine the correlation length from properties of two-particle states, not one-particle states, and we regard our values for ξ as potentially unreliable (therefore the symbol $\approx$ in table 2). Nevertheless, we can only underestimate ξ, so our conclusions regarding critical slowing down are not spoiled.

The H-algorithm was not successful in eliminating CSD in either of the two regimes. The exponential autocorrelation times displayed in tables 1 and 2 suggest that z is around 1 at $\beta=1$ (from the two largest lattices), and consistent with 2 in the massive phase (recall that $z=2$ or slightly above is typical for local algorithms, which strongly suffer from CSD [8]). However, we should be careful in drawing a definite conclusion concerning the value of z, since at $\beta=1$ there are strong deviations from the L^z behaviour (for $L=64$), and for the $\xi<L$ runs we considered only two different lattice sizes.

A careful analysis of the cluster size distribution revealed that the H-algorithm produces both small and very large clusters quite frequently, while intermediate-size clusters are comparatively rare. Nearly half of the work is spent in clusters larger than 90% of the lattice, which, similarly to the Ising model studies, do not considerably change a configuration. This situa-

Table 1
Exponential autocorrelation times for $\beta=1$ ($\xi=\infty$).

L	Algorithm			
	H	IH	QH	QM
16	13.5(3.5)	-	-	-
32	60(20)	7(1)	-	-
64	82(18)	8(1)	12.5(1.5)	26.5(4.5)
128	165(33)	11.5(1.5)	13(1.5)	58(12)
256	-	11(2)	-	-

Table 2
Exponential autocorrelation times for $\beta<\beta_c$.

β	ξ	L	Algorithm	
			H	IH
0.65	≈ 14	64	12.3(1.5)	9(2)
0.65	≈ 14	128	-	9(3)
0.68	≈ 44	64	-	8(2)
0.68	≈ 44	128	41.5(7.0)	11(2)
0.68	≈ 44	256	-	13(2)

[#1] Since the discrete gaussian model is the large fugacity limit of the sine-Gordon model, the kinds discussed here are nothing but the well known kinks of the renormalizable phase of that model.

tion can be understood if we take into account the fact that for the couplings and lattice sizes considered in our study, the surface thickness is rather small: at $\xi=20$ it is less than 1; in the massless phase it increases logarithmically with L, but at $\beta=1$ for $L=256$ it is still only 1.4. As a consequence, the reflection point M often lies above the top of a hill or below the bottom of a valley, in contradistinction with our original intuitive picture.

A natural attempt to improve the situation is to include reflections with respect to integer valued planes M. In order to get clusters of sizes larger than one site (which cannot happen when M equals the seed spin m_{x0} and $q=1$), but still assure that the reflection plane lies frequently enough within the (rather narrow) vertical bounds of the hill-and-valley landscape, we devised the following two algorithms:

Algorithm Q. $M=m_{x0}$, i.e. the the reflection plane equals the seed spin, but $q=q_0$ for $|m_x-M| \times |m_y-M|=0$ with q_0 some constant strictly smaller than 1, and $q=1$ otherwise.

Algorithm I. $M=m_{y0}$, where y_0 is a randomly chosen lattice site which is different from the seed x_0, and $q=1$ always.

Both these algorithms are non-ergodic, since they only change the spins by even amounts. Therefore they have to be combined with other procedures. We studied the combinations IH and QH of the I and Q algorithms with the H-algorithm.

Notice that for the Q-algorithm, a cluster grown from the seed x_0 may contain spins that are above, below and equal to M. This is a quite unusual situation in the context of embedded Ising variables, but is perfectly allowed within the framework for cluster algorithms that we used here [3].

From the values of τ displayed in table 1, one can conclude that the value of z for the IH algorithm is very small, possibly zero, at $\beta=1$. The results for $\beta<\beta_c$ also suggest a small value of z, but more data are necessary to reach a definite conclusion (see table 2). Thus the IH algorithm turned out to be *extremely efficient in eliminating* CSD. Furthermore, the results in table 1 show that there is no significant difference between the performance of the IH algorithm and that of the QH algorithm.

By varying the ratio of the number of I and H clusters (within reasonable bounds) the autocorrelation times did not change significantly. In tables 1 and 2 we only presented the results of runs where this ratio was one-to-one. Similarly, changing the value of q_0 such that the mean cluster size ranged between a quarter and a third of the lattice (q_0 was around 0.7) did not have a significant impact on the performance of the QH algorithm.

The IH and QH algorithms generate considerably more medium-size clusters and less large clusters than the H-algorithm alone. This confirms our expectation that choosing the reflection plane equal to one of the spins, not at a distance of $\pm\frac{1}{2}$ away, is very helpful in the case of a thin SOS surface. However, the good performance of these algorithms is not a function of the cluster size distribution alone, it is also sensitive to other details of the configuration changes that occur when the various types of clusters are flipped.

Let us attempt an explanation of why the IH and QH algorithms work so well.

4. Why does it work?

At small β the important configurations consist of a large flat surface, with a few two-dimensional regions that are *one* unit higher or lower. We shall call these regions *single-step-islands* (SSI). In hamiltonian (transfer matrix) language, the SSI's correspond to the surface between the world lines of a kink–antikink pair (remember that the kinks are the small β particle excitations, see footnote 1). As β_c is approached, SSI's become larger, more frequent, and more often on top of one another. Their condensation causes the SOS surface roughening at β_c.

If we have an SSI on top of a flat background of height M, there is a large probability that the I or Q algorithms reflect it with respect to that background. Since the original and the reflected configuration have the same Boltzmann weight, the reflection of the SSI is a kind of microcanonical move. Such low-energy-cost large-scale changes in the configurations are usually very efficient in decreasing the autocorrelation time. We believe that this is the main reason why the cluster flips with respect to an integer valued reflection plane M improve the situation so dramatically.

The I and Q algorithms have to be combined with another algorithm in order to ensure ergodicity. We now ask the question whether, in order to overcome

CSD, it is crucial to combine I or Q with the H-algorithm, or whether we may also use a local ergodic algorithm. If M is an integer, no large SSI's can be created or destroyed by cluster reflections. The H-algorithm on the other hand can achieve this easily. Of course, a local algorithm like Metropolis cannot create or destroy any large scale objects. We may therefore expect that a combination IM or QM of I or Q with a Metropolis procedure will exhibit CSD. The study of the QM algorithm, whose results are also shown in table 1, clearly shows that this is indeed what happens.

For the QM algorithm we did one Metropolis sweep for, roughly, one work unit of the cluster part. The values of τ quoted in table 1 disregard the contribution of the Metropolis sweeps to the total amount of work. Even if we weigh the work done by the two procedures equally (which increases the values of τ by a factor of roughly 2) the QM algorithm seems to be more efficient than the H-algorithm.

We conclude that our picture of the SSI's as the relevant objects for understanding our cluster algorithms is correct, and that is absolutely crucial to use both the integer *and* half-integer reflection planes.

5. Conclusions

We described a new class of cluster algorithms which eliminate critical slowing down for SOS models. We also provided the physical intuition for understanding why this kind of algorithms are successful.

It is remarkable that the successful algorithms are mixtures of a slow algorithm (the H-algorithm) with a non-ergodic algorithm (I or Q), which by itself is also slow in the sense that supplementing it with a local ergodic algorithm is not sufficient to overcome CSD.

We would like to point out that our algorithms can also be applied to other SOS-type models. If in (1) $(m_x-m_y)^2$ is generalized to some function of $|m_x-m_y|$, our simulation methods apply almost without any modification. Generalizations to restricted SOS models are also possible.

Furthermore, the type of cluster algorithms we described can be adapted for continuous-spin models (two-dimensional scalar field theories), like the sine-Gordon model, the dipole gas, and the massless free field theory. Actually, for the massless free field theory and for the sine-Gordon model we already have results that show (almost) no critical slowing down.

For the integer M part of our algorithm, we expect a Swendsen–Wang multicluster procedure [1] to perform similarly well to the single cluster procedure, one reason being that no work at all would be performed in connection to the spins equal to M.

Detailed numerical investigations of the KT transition in SOS models and in the sine-Gordon model are in progress.

In ref. [9] a related surface-cluster algorithm is described for the interface of the three-dimensional Ising model with mixed boundary conditions.

A study of our algorithms for the three-dimensional discrete gaussian model is also in progress. The results obtained so far are much less spectacular than in two dimensions. This is an indication that the relevant excitations of the model (long-wavelength spin waves) are not as effectively updated as the SSI's of the SOS model. On the other hand, we have very encouraging results for the massless free field theory in three dimensions. This indicates that cluster algorithms of the type described may be efficient also in the case of spin waves, and that it is worth while to continue the investigations in three dimensions. The most promising path to be pursued in this context seems to be a combination of the cluster algorithm with a multigrid procedure.

Acknowledgement

We wish to thank R. Ben-Av, D. Kandel and G. Mack for valuable discussions. One of us (M.M.) gratefully acknowledges several discussions with A. Sokal. Three of us (H.G.E., M.M. and K.P.) have enjoyed the kind hospitality of the Weizmann Institute and the Hebrew University. Support by the Deutsche Forschungsgemeinschaft is gratefully acknowledged by H.G.E., M.H. and K.P. This work was also supported in part by grants No. I-131-095.07/89 from the German–Israeli Foundation for Research and Development (GIF) and No. 517/89 from the Basic Research Foundation of The Israel Academy of Sciences and Humanities. Last but not least we would like to express our gratitude to the HLRZ in Jülich, were our computer runs were performed.

References

[1] R.H. Swendsen and J.S. Wang, Phys. Rev. Lett. 58 (1987) 86;
R.C. Brower and P. Tamayo, Phys. Rev. Lett. 62 (1989) 1087.

[2] U. Wolff, Phys. Rev. Lett. 62 (1989) 361; Nucl. Phys. B 322 (1989) 759; Phys. Lett. B 228 (1989) 379.

[3] D. Kandel et al., Phys. Rev. Lett. 60 (1988) 1591;
D. Kandel, E. Domany and A. Brandt, Phys. Rev. B 40 (1989) 330;
D. Kandel and E. Domany, Weizmann Institute preprint (September 1990).

[4] A.D. Sokal, Monte Carlo methods in statistical mechanics: foundations and new algorithms, Cours de Troisième Cycle de la Physique en Suisse Romande (Lausanne, June 1989).

[5] For a nice discussion see: R.H. Swendsen, Phys. Rev. B 18 (1978) 492;
For reviews on SOS type of models see: D.B. Abraham, Surface structures and phase transitions – exact results in: Phase transitions and critical phenomena, Vol. 10, eds. C. Domb and J.L. Leibowitz (Academic Press, New York, 1986);
H. van Beijeren and I. Nolden, The roughening transition, in: Topics in current physics, Vol. 43, eds. W. Schommers and P. van Blankenhagen (Springer, Berlin, 1987).

[6] J.M. Kosterlitz and D.J. Thouless, J. Phys. C 6 (1973) 1181;
J.M. Kosterlitz, J. Phys. C 7 (1974) 1046.

[7] R.G. Edwards, J. Goodman and A.D. Sokal, preprint FSU-SCRI-90-99 (June 1990).

[8] P.C. Hohenberg and B.I. Halperin, Rev. Mod. Phys. 49 (1977) 435.

[9] M. Hasenbusch and S. Meyer, Cluster update acceleration of interface roughening in the 3D Ising model, Kaiserslautern preprint TH 14-90 (September 1990).