

## HOW TO BEAT CRITICAL SLOWING-DOWN: 1990 UPDATE

Alan D. SOKAL

Department of Physics, New York University, 4 Washington Place, New York, NY 10003 USA  
 (SOKAL@ACF4.NYU.EDU)

We discuss the problem of critical slowing-down (CSD) in Monte Carlo simulations, and review recent progress in devising collective-mode algorithms that reduce or eliminate CSD. Emphasis is on cluster and embedding algorithms; multigrid and Fourier acceleration are also discussed briefly.

### 1. GENERAL THEORY

This talk is the latest installment in an annual tradition of reviews on the problem of critical slowing-down in Monte Carlo simulations [1,2,3,4]. In particular, much useful background can be found in last year's review by Ulli Wolff [4] and in the previous year's review by Steve Adler [3]. For lack of space-time I will *not* discuss here: algorithms for dynamical fermions [5]; algorithms for computing propagators [6,7]; or algorithms for geometrical problems such as self-avoiding walks [8], random graphs [9] and random surfaces.

So what is critical slowing-down? Let us recall the general principles of dynamic Monte Carlo methods [10,11]: We wish to generate random samples from some probability distribution  $\pi(\varphi)$  [usually a Gibbs measure  $e^{-H(\varphi)}$ ]. To do this, we invent a transition probability matrix  $P(\varphi \rightarrow \varphi')$  that is ergodic ("from any state one can reach any other") and that leaves  $\pi$  invariant:

$$\int d\varphi \pi(\varphi) P(\varphi \rightarrow \varphi') = \pi(\varphi'). \quad (1)$$

[The detailed-balance condition  $\pi(\varphi) P(\varphi \rightarrow \varphi') = \pi(\varphi') P(\varphi' \rightarrow \varphi)$  is sufficient but not necessary.] We then simulate the Markov process defined by  $P$ , starting in some arbitrary initial configuration  $\varphi^{(0)}$ ; this generates a random sequence of configurations  $\varphi^{(1)}, \varphi^{(2)}, \dots, \varphi^{(n)}$ , where  $n$  is the run length. After an initial transient (which we discard), the system will have settled down to very nearly the unique equilibrium probability distribution  $\pi$ . So

\*To some extent, in some favorable cases. (Albeit probably not the cases *you* want to study.)

we form sample means  $\bar{A} \equiv n^{-1} \sum_{i=1}^n A(\varphi^{(i)})$ , and take them as estimates of the theoretical means  $\langle A \rangle \equiv \int d\varphi \pi(\varphi) A(\varphi)$ .

The trouble, of course, is that the samples  $\varphi^{(1)}, \varphi^{(2)}, \dots$  are not statistically independent, but are in general highly correlated. If the "autocorrelation time" of the Markov chain is  $\tau$ , then a run of length  $n$  provides only  $\sim n/\tau$  "effectively independent" samples. In particular, near a critical point (second-order phase transition),  $\tau$  typically diverges as

$$\tau \sim \min(L, \xi)^z, \quad (2)$$

where  $L$  is the lattice size,  $\xi$  is the correlation length in an infinite-volume system at the same parameters, and  $z$  is a *dynamic critical exponent*. For the traditional local algorithms — such as the single-site Metropolis and heat-bath algorithms, the unaccelerated Langevin algorithm, and so forth —  $z$  is typically near 2 (see Section 2). This divergence of  $\tau$  at a critical point is called *critical slowing-down* (CSD). For a theorist studying dynamic critical behavior, this is a fascinating phenomenon; but for a practitioner of Monte Carlo, it is a pain. The goal of much research over the last five years has been, therefore, to invent new Monte Carlo algorithms with radically reduced critical slowing-down.

(Let me remark that near a first-order transition the slowing-down is even more severe: typically  $\tau \sim \exp(cL^{d-1})$ , as required for tunneling through very improbable configurations involving interfaces. Almost no progress has been made, to my knowledge, in devising algorithms that alleviate first-order slowing-down.)

It is necessary to be a bit more precise: there are at least two distinct definitions of  $\tau$ , and correspondingly, two distinct exponents  $z$ . Consider an observable  $A$ , and let

$$C_{AA}(t) = \langle A^{(s)} A^{(s+t)} \rangle - \langle A \rangle^2 \quad (3)$$

be its unnormalized autocorrelation function in the stationary Markov process (i.e. in equilibrium). Define also the normalized autocorrelation function

$$\rho_{AA}(t) = C_{AA}(t)/C_{AA}(0). \quad (4)$$

Typically  $\rho_{ff}(t)$  decays exponentially ( $\sim e^{-t|\tau|}$ ) for large  $t$ ; we define the exponential autocorrelation times

$$\tau_{exp,A} \equiv \limsup_{t \rightarrow \infty} \frac{t}{-\log |\rho_{AA}(t)|} \quad (5)$$

$$\tau_{exp} \equiv \sup_A \tau_{exp,A} \quad (6)$$

Thus,  $\tau_{exp,A}$  is the relaxation time of the slowest mode which couples to  $A$ , and  $\tau_{exp}$  is the relaxation time of the slowest mode in the system. (For most observables,  $\tau_{exp,A} = \tau_{exp}$ ; the exception is when  $A$  is "orthogonal" to the slowest mode.) On the other hand, the statistical error in estimates of  $\langle A \rangle$  is controlled by the integrated autocorrelation time

$$\tau_{int,A} \equiv \frac{1}{2} \sum_{t=-\infty}^{\infty} \rho_{AA}(t), \quad (7)$$

in the sense that

$$\text{var} \left( \frac{1}{n} \sum_{t=1}^n A^{(t)} \right) \approx \frac{2\tau_{int,A}}{n} C_{AA}(0). \quad (8)$$

In other words, a run of length  $n$  contains  $n/2\tau_{int,A}$  "effectively independent" samples. (This is sometimes expressed by saying that the "statistical inefficiency" of dynamic Monte Carlo, relative to static Monte Carlo, is  $2\tau_{int,A}$ .) In summary, the exponential and integrated autocorrelation times play different roles:  $\tau_{exp}$  is very natural from the point of view of the theory of dynamic critical phenomena, while  $\tau_{int,A}$  is of practical importance.

It seems to be generally believed that  $\tau_{exp}$  and  $\tau_{int,A}$  are of the same order of magnitude, i.e. diverge with the same dynamic critical exponent  $z$ . This belief is implicit in articles which refer simply to "the" dynamic critical exponent  $z$ ; and it is

made explicit in some of my own papers of a few years back (which in my embarrassment I refrain from citing). It now seems to me quite obvious that  $\tau_{exp}$  and  $\tau_{int,A}$  need not be of the same order of magnitude, i.e. they need not scale with the same dynamic critical exponent  $z$ . So we should define distinct dynamic critical exponents  $z_{exp}$  and  $z_{int,A}$ .\*

$$\tau_{exp,A} \sim \tau_{exp} \sim \min(L, \xi)^{z_{exp}} \quad (9)$$

$$\tau_{int,A} \sim \min(L, \xi)^{z_{int,A}} \quad (10)$$

Nearly always one has  $\tau_{int,A} \lesssim \tau_{exp,A}$  (this is provable if detailed balance holds [11]), hence  $z_{int,A} \leq z_{exp}$ . But  $z_{int,A}$  can be strictly smaller than  $z_{exp}$ ! One known example is the pivot algorithm for the ordinary random walk, which is exactly soluble [12, Section 3.3]. But I claim that in fact  $z_{int,A} < z_{exp}$  should be regarded as the typical behavior!

To see this, consider the following analogies between dynamic and static critical phenomena:

<u>Dynamic</u>	<u>Static</u>
time $\longleftrightarrow$	space
$\tau_{exp} \longleftrightarrow$	$\xi$
$\tau_{int,A} \longleftrightarrow$	$\left\{ \begin{array}{l} \text{susceptibility } \chi \text{ (if } A = \mathcal{M}) \\ \text{specific heat } C_h \text{ (if } A = \mathcal{E}) \\ \text{etc.} \end{array} \right.$

Now we know perfectly well that the susceptibility and correlation length have different critical exponents ( $\gamma \neq \nu$ ); so shouldn't one expect that the integrated and exponential autocorrelation times do likewise ( $z_{int,A} \neq z_{exp}$ )? Indeed, there is a scaling law relating  $\gamma$  to  $\nu$  and the exponent  $\eta$  describing the decay of correlations at criticality; and one expects a similar scaling law for dynamic correlations:

$$\begin{array}{ccc} \rho_{AA}(t) \sim t^{-PA} F(t/\tau_{exp}) & \longleftrightarrow & G_{AA}(x) \sim x^{-(d-2+\eta_A)} F(x/\xi) \\ \Downarrow & & \Downarrow \\ z_{int,A} = (1-PA)z_{exp} & & \gamma_A = (2-\eta_A)\nu \end{array}$$

\*Even for observables  $A$  that are orthogonal to the slowest mode, one typically expects that  $\tau_{exp,A} \sim \tau_{exp}$ . For example, in any linear stochastic iteration for a Gaussian model, the observables  $\varphi^n$  have an exponential autocorrelation time which is exactly  $1/n$  times that of the slowest mode [14, Section VIII].

So one expects  $z_{\text{ind},A} < z_{\text{exp}}$  except in the special case when  $p_A = 0$ . This latter does occur in some simple examples (e.g. in most algorithms for the Gaussian model), but I see no reason for it to occur even in the Glauber dynamics for the two-dimensional Ising model! The moral is: (1) the theory of dynamic critical phenomena needs to be rethought; (2) future numerical papers need to distinguish between  $z_{\text{ind},A}$  and  $z_{\text{exp}}$  (and to specify  $A!$ ); and (3) past numerical papers need to be reanalyzed.

## 2. COLLECTIVE-MODE MONTE CARLO

The critical slowing-down of the conventional Monte Carlo algorithms arises fundamentally from the fact that their updates are *local*: in a single step of the algorithm, "information" is transmitted from a given site or link only to its nearest neighbors. Crudely one might guess that this "information" executes a random walk around the lattice. In order for the system to evolve to an "essentially new" configuration, the "information" has to travel a distance of order  $\xi$ , the (static) correlation length. One would guess, therefore, that  $\tau \sim \xi^2$  near criticality, i.e. that the dynamic critical exponent  $z$  equals 2. This guess is correct for the Gaussian model (free field).<sup>†</sup> For other models, we have a situation analogous to theory of static critical phenomena: the dynamic critical exponent is a nontrivial number that characterizes a rather large class of algorithms (a so-called "dynamic universality class"). In any case, for most models of interest, the dynamic critical exponent for local algorithms is close to 2 (usually somewhat higher) [15,16]. Accurate measurements of dynamic critical exponents are, however, very difficult — even more difficult than measurements of static critical exponents — and require enormous quantities of Monte Carlo data: run lengths of  $\gtrsim 10000\tau$ , when  $\tau$  is itself getting large!<sup>‡</sup>

We can now make a rough estimate of the com-

puter time needed to study the Ising model near its critical point, or quantum chromodynamics near the continuum limit. Each sweep of the lattice takes a time of order  $L^d$ , where  $d$  is the spatial (or space-"time") dimensionality of the model. And we need  $\approx 2\tau$  sweeps in order to get one "effectively independent" sample. So this means a computer time of order  $L^d \xi^z \gtrsim \xi^{d+z}$ .<sup>§</sup> For high-precision statistics one might want  $10^6$  "independent" samples. The reader is invited to plug in  $\xi = 100$ ,  $d = 4$  (or  $d = 3$  if you're a condensed-matter physicist) and get depressed. It should be emphasized that the factor  $\xi^d$  is inherent in *all* Monte Carlo algorithms for spin models and field theories (but not for self-avoiding walks [1,2,8,11,19]). The factor  $\xi^z$  could, however, conceivably be reduced or eliminated by a more clever algorithm.

What is to be done? Our knowledge of the physics of critical slowing-down tells us that the slow modes are the long-wavelength modes, if the updating is purely local. The natural solution is therefore to speed up those modes by some sort of *collective-mode* (nonlocal) updating. It is necessary, then, to *identify physically the appropriate collective modes*, and to *devise an efficient computational algorithm* for speeding up those modes. These two goals are unfortunately in conflict: it is very difficult to devise collective-mode algorithms that are not so nonlocal that their increased computational complexity per iteration outweighs the reduction in critical slowing-down. (For example, in  $d = 4$ , an algorithm that eliminates critical slowing-down but has computational complexity  $O(V^2)$  is as bad as an  $O(V)$  algorithm whose dynamic critical exponent is  $z = 4$  — i.e. much *worse* than the conventional algorithms.)

Specific implementations of the collective-mode idea are thus highly model-dependent: one has to use one's knowledge of the *physics* of a given model to build into the algorithm the collective modes that *the system wants* — and do

<sup>†</sup>Indeed, for the Gaussian model this random-walk picture can be made rigorous: see [13] combined with [14, Section 8].

<sup>‡</sup>The statistical aspects of such studies are discussed in [11], [12, Appendix C] and [17].

<sup>§</sup>Clearly one must take  $L \gtrsim \xi$  in order to avoid severe finite-size effects. Typically one approaches the critical point with  $L \approx c\xi$ , where  $c \approx 2 - 4$ , and then uses finite-size scaling [18] to extrapolate to the infinite-volume limit.

so at not too great a computational cost. Three classes of collective-mode algorithms have been invented so far, and found to be advantageous in at least some models:

- Fourier acceleration [20]
- Multigrid Monte Carlo (MGMC) [21,14,22]
- Auxiliary-variable (cluster) algorithms [23,24,25]

(Overrelaxation and hybrid are not collective-mode algorithms, but they can in some cases reduce the critical slowing-down from  $z \approx 2$  to  $z \approx 1$  [3,26].)

### 3. FOURIER ACCELERATION AND MULTIGRID

Fourier acceleration and multigrid Monte Carlo are discussed in detail in references [20] and [14], respectively. Here I want to make only a brief comparison.

Fourier acceleration and multigrid are philosophically and physically very similar (though their technical details are quite different). Both are based on an intuition from the free-field (Gaussian) model, and both can be proven to eliminate completely the critical slowing-down in this model.<sup>¶</sup> Both offer the system collective-mode updates of *fixed shape* (sine waves in Fourier acceleration, typically square or triangular waves in multigrid) on *all length scales*, and allow the system to choose the amplitude. (These algorithms make sense, therefore, only for systems of continuous-valued spins.) Both algorithms are expected to work well (i.e. have  $z \approx 0$ ) for systems that are in some sense near-Gaussian, such as asymptotically free continuous-spin models or the low-temperature (spin-wave) phase of the two-dimensional  $XY$  model. Both algorithms are expected to work badly (i.e. have

<sup>¶</sup>This is not, of course, such a great feat: the critical slowing-down in the Gaussian model can also be eliminated by throwing away the computer and solving the model analytically. I consider the Gaussian model, rather, as a constraint: if an algorithm does not work well for the Gaussian model, then it is unlikely to work well for near-Gaussian models (e.g. asymptotically free theories) either.

$z \approx 2$ ) for systems in which the dominant large-scale collective modes have discrete elements, such as spin-flips in the one-component  $\varphi^4$  model or vortices in the two-dimensional  $XY$  model near the Kosterlitz-Thouless transition. In summary, the performance of Fourier acceleration and MGMC is probably very similar, in the sense that they probably work well for the same models and work badly for the same models; it is even conceivable that in many models they are in the same dynamic universality class.

Two recent studies of MGMC for the two-dimensional  $XY$  model [22,27] confirm these predictions: they find  $z_{MGMC} \approx 1.4$  as the critical temperature is approached from above, compared to  $z_{heat-bath} \approx 2.1$ ; but they find  $z_{MGMC} = 0$  in the low-temperature (spin-wave) phase. Studies of MGMC for asymptotically free  $\sigma$ -models are now in progress [28,29]. Correspondingly detailed studies of the dynamic critical behavior of the Fourier-accelerated Langevin and hybrid algorithms have not yet been done — but they ought to be! (Whoever does them should use the exact versions of these algorithms [30,31].)

### 4. AUXILIARY-VARIABLE (CLUSTER) ALGORITHMS

A very different type of collective-mode algorithm was proposed three years ago by Swendsen and Wang [23] for Potts spin models. Since then, there has been an explosion of work trying to understand why this algorithm works so well and why it does not work even better, and trying to improve or generalize it. The basic idea behind all algorithms of Swendsen-Wang type is to augment the given model by means of *auxiliary variables*, and then to simulate this augmented model. (Algorithms of this type are sometimes called “cluster algorithms”, but this term is too narrow, because the relevant objects are not always clusters — see below.)

Consider the Hamiltonian for the ferromagnetic  $q$ -state Potts model [32]:

$$H(\sigma) = - \sum_{\langle ij \rangle} J_{ij} (\delta_{\sigma_i, \sigma_j} - 1), \quad (11)$$

where  $J_{ij} \geq 0$  for all  $i, j$ . The partition function is

then

$$Z = \sum_{\{\sigma\}} \exp \left[ \sum_{\langle ij \rangle} J_{ij} (\delta_{\sigma_i, \sigma_j} - 1) \right] \\ = \sum_{\{\sigma\}} \prod_{\langle ij \rangle} [(1 - p_{ij}) + p_{ij} \delta_{\sigma_i, \sigma_j}] \quad (12)$$

where we have defined  $p_{ij} = 1 - \exp(-J_{ij})$ . We now employ the deep identity

$$a + b = \sum_{n=0}^1 [a \delta_{n,0} + b \delta_{n,1}], \quad (13)$$

where  $a$  and  $b$  are real numbers.<sup>||</sup> That is, we introduce on each bond  $\langle ij \rangle$  an auxiliary variable  $n_{ij}$  taking the values 0 and 1, and obtain

$$Z = \sum_{\{\sigma\}} \sum_{\{n\}} \prod_{\langle ij \rangle} [(1 - p_{ij}) \delta_{n_{ij},0} + p_{ij} \delta_{n_{ij},1} \delta_{\sigma_i, \sigma_j}]. \quad (14)$$

Let us now take seriously the  $\{n\}$  as dynamical variables: we can think of  $n_{ij}$  as an *occupation variable* for the bond  $\langle ij \rangle$  (1 = occupied, 0 = empty). We therefore define the *Fortuin-Kasteleyn-Swendsen-Wang (FKSW) model* to be a joint model having  $q$ -state Potts spins  $\sigma_i$  at the sites and occupation variables  $n_{ij}$  on the bonds, with the joint probability distribution implied by (14). Finally, let us see what happens if we sum over the  $\{\sigma\}$  at fixed  $\{n\}$ . Each occupied bond  $\langle ij \rangle$  imposes a constraint that the spins  $\sigma_i$  and  $\sigma_j$  must be in the same state, but otherwise the spins are unconstrained. We therefore group the sites into connected clusters (two sites are in the same cluster if they can be joined by a path of occupied bonds); then all the spins within a cluster must be in the same state (all  $q$  values are equally probable), and distinct clusters are independent. It follows that

$$Z = \sum_{\{n\}} \left( \prod_{\langle ij \rangle: n_{ij}=1} p_{ij} \right) \left( \prod_{\langle ij \rangle: n_{ij}=0} (1 - p_{ij}) \right) q^{C(n)}, \quad (15)$$

where  $C(n)$  is the number of connected clusters (including one-site clusters) in the graph whose edges are the bonds having  $n_{ij} = 1$ . The corresponding

<sup>||</sup>This identity is valid in an arbitrary abelian semigroup, but such generality will not be needed here.

probability distribution is called the *random-cluster model with parameter  $q$*  [36]: it is a generalized bond-percolation model with non-local correlations coming from the factor  $q^{C(n)}$ , and for  $q = 1$  it reduces to ordinary bond percolation.

We have thus verified the following facts about the FKSW model:

- a)  $Z_{\text{Potts}} = Z_{\text{FKSW}} = Z_{\text{RC}}$ .
- b) The marginal distribution of  $\mu_{\text{FKSW}}$  on the Potts variables  $\{\sigma\}$  (integrating out the  $\{n\}$ ) is precisely the Potts model  $\mu_{\text{Potts}}(\sigma)$ .
- c) The marginal distribution of  $\mu_{\text{FKSW}}$  on the bond occupation variables  $\{n\}$  (integrating out the  $\{\sigma\}$ ) is precisely the random-cluster model  $\mu_{\text{RC}}(n)$ .

The conditional distributions of  $\mu_{\text{FKSW}}$  are also simple:

- d) The conditional distribution of the  $\{n\}$  given the  $\{\sigma\}$  is as follows: independently for each bond  $\langle ij \rangle$ , one sets  $n_{ij} = 0$  in case  $\sigma_i \neq \sigma_j$ , and sets  $n_{ij} = 0, 1$  with probability  $1 - p_{ij}, p_{ij}$ , respectively, in case  $\sigma_i = \sigma_j$ .
- e) The conditional distribution of the  $\{\sigma\}$  given the  $\{n\}$  is as follows: independently for each connected cluster, one sets all the spins  $\sigma_i$  in the cluster to the same value, chosen equiprobably from  $\{1, 2, \dots, q\}$ .

The Swendsen-Wang (SW) algorithm [23] simulates the joint model (14) by alternately applying the conditional distributions (d) and (e) — that is, by alternately generating new bond occupation variables (independent of the old ones) given the spins, and new spin variables (independent of the old ones) given the bonds. Each of these operations can be carried out in a computer time of order volume: for generating the bond variables this is trivial, and for generating the spin variables it relies on an efficient (linear-time) algorithm for computing the connected clusters [33,34,35]. It is easy to see that the SW algorithm is ergodic and leaves

$L$	$\chi$	$\tau_{\text{int},\mathcal{E}}(\text{SW})$	$\tau_{\text{exp},\mathcal{M}}(\text{Metr})$
64	1575 ( 10)	5.25 (0.30)	5380 (140)
128	5352 ( 53)	7.05 (0.67)	23950 (480)
256	17921 (109)	6.83 (0.40)	<i>104500 (est.)</i>
512	59504 (632)	7.99 (0.81)	<i>458000 (est.)</i>

Table 1: Data for two-dimensional Ising model at criticality. Susceptibility  $\chi$  and Swendsen-Wang autocorrelation time  $\tau_{\text{int},\mathcal{E}}$  ( $\mathcal{E}$  = energy  $\approx$  slowest mode) are from [2]. Metropolis autocorrelation time  $\tau_{\text{exp},\mathcal{M}}$  ( $\mathcal{M}$  = magnetization  $\approx$  slowest mode) is from [39]; data in italics are extrapolations. Standard error is shown in parentheses.

invariant the distribution (14).\*\*

It is certainly plausible that the SW algorithm might have less critical slowing-down than the conventional (single-spin-update) algorithms: the reason is that a local move in one set of variables can have highly nonlocal effects in the other. For example, setting  $n_b = 0$  on a single bond may disconnect a cluster, causing a big subset of the spins in that cluster to be flipped simultaneously. In some sense, therefore, the SW algorithm is a collective-mode algorithm in which the collective modes are *chosen by the system* rather than imposed from the outside as in Fourier acceleration or multigrid. (The miracle is that this is done in a way that preserves the correct Gibbs measure.)

How well does the SW algorithm perform? Table 1 shows some data [2] on a two-dimensional Ising model at the bulk critical temperature; for comparison we give also data on the single-site Metropolis algorithm [39]. These data are con-

\*\**Historical remark:* The random-cluster model was introduced in 1969 by Fortuin and Kasteleyn [36]; they derived the identity  $Z_{\text{Potts}} = Z_{\text{RC}}$  and some corresponding identities for correlation functions. These relations were rediscovered several times during the subsequent two decades [37]. Surprisingly, however, no one seems to have noticed the *joint* probability distribution (14) that underlay all these identities; this was discovered implicitly by Swendsen and Wang [23], and was made explicit by Edwards and Sokal [38].

	$q = 1$	$q = 2$	$q = 3$	$q = 4$
$d = 1$	0	0	0	0
$d = 2$	0	$0 \times \log (?)$ $\approx 0.35 (?)$	$0.55 \pm 0.03$	$\approx 1$ (exact?)
$d = 3$	0	$\approx 0.34 (?)$ $\approx 0.75 (?)$	—	—
$d = 4$	0	1 (exact?)	—	—

Table 2: Current best estimates of the dynamic critical exponent  $z$  for the Swendsen-Wang algorithm. References:  $d = 2, q = 2$  [40,41,23];  $d = 2, q = 3$  [42,41];  $d = 2, q = 4$  [42];  $d = 3, q = 2$  [43,17,23];  $d = 4, q = 2$  [44]. Error bar is a 95% confidence interval.

sistent both with  $\tau_{\text{SW}} \sim L^{\approx 0.35}$  [23] and with  $\tau_{\text{SW}} \sim \log L$  [40] (it seems difficult to distinguish except by using *extremely* large lattices, e.g.  $L$  up to 2000 or more). By contrast, the Metropolis algorithm has  $z \approx 2.13$  [39]. For  $L = 512$ , this translates into a factor-of-50000 advantage for SW over Metropolis. Even granting that one iteration of the Swendsen-Wang algorithm may be a factor of  $\sim 10 - 100$  more costly in CPU time than one iteration of a conventional algorithm (the exact factor depends on the efficiency of the cluster-finding subroutine), the SW algorithm wins already for  $L \gtrsim 25$ .

For other Potts models, the performance of the SW algorithm is less spectacular for the two-dimensional Ising model, but it is still very impressive. In Table 2 we give the current best estimates of the dynamic critical exponent  $z_{\text{SW}}$  for  $q$ -state Potts models in  $d$  dimensions, as a function of  $q$  and  $d$ . (For the SW algorithm the decay of the energy-energy autocorrelation function appears to be very close to exponential, hence  $z_{\text{int},\mathcal{E}} \approx z_{\text{exp}}$ .) All these exponents are much lower than the  $z \gtrsim 2$  observed in the single-spin-flip algorithms.

Although the SW algorithm performs extraordinarily well, we understand very little about *why* these exponents take the values they do. Some cases are easy. If  $q = 1$ , then all spins are in the same state (the *only* state!), and all bonds are

thrown independently, so the autocorrelation time is zero. (Here the SW algorithm just reduces to the standard *static* algorithm for independent bond percolation.) If  $d = 1$  (more generally, if the lattice is a *tree*), the SW dynamics is exactly soluble: the behavior of each bond is independent of each other bond, and  $\tau_{exp} \rightarrow -1/\log(1 - 1/q) < \infty$  as  $\beta \rightarrow +\infty$ . But the remainder of our understanding is very murky. Two principal insights have been obtained so far:

- a) A calculation yielding  $z_{SW} = 1$  in a mean-field (Curie-Weiss) Ising model [45]. This suggests (but of course does not prove) that  $z_{SW} = 1$  for Ising models ( $q = 2$ ) in dimension  $d \geq 4$ .
- b) A rigorous proof that  $z_{SW} \geq \alpha/\nu$  [42]; physically this is due to the slow convergence of energy-like observables. This bound, while valid for all  $d$  and  $q$ , is extremely far from sharp for the Ising models in dimensions 3 and higher. But it is reasonably good for the 3- and 4-state Potts models in two dimensions, and in the latter case it may even be sharp.

But much remains to be understood!

Numerous modifications and generalizations of the SW algorithm have been proposed. For Potts and related models we have:

- Single-cluster variant [24]
- Duality-improved SW algorithm ( $d = 2$  only) [47]
- Multi-scale SW algorithms [51]
- SW algorithm for Potts lattice gauge theories [49,50]
- Algorithm for fully frustrated Ising models [52]

For non-Potts models, several generalizations have been proposed [38,53], but the most promising ideas at present seem to be the *embedding algorithms* discussed in the next section.

In the single-cluster (1C) variant of the SW algorithm [24], one builds only a *single* cluster (starting at a randomly chosen site) and flips it — as

opposed to the standard SW algorithm, which enumerates *all* the clusters in the lattice. Clearly, one step of the single-cluster SW algorithm makes less change in the system than one step of the standard SW algorithm, but it also takes much less work; what matters is the dynamic critical exponent  $z$  measured in CPU-time units. One advantage of the single-cluster algorithm is that the probability of choosing a cluster is proportional to its size (since we pick a random site), so the work is concentrated preferentially on larger clusters. Another advantage is that the successive clusters are less strongly correlated. So it would not be surprising if  $z_{1C,CPU}$  were smaller than  $z_{SW}$ . The measurements thus far [17,41,46] paint a confusing picture:  $z_{1C,CPU}$  and  $z_{SW}$  seem to be roughly the same in  $d = 2$ ;  $z_{1C,CPU}$  seems to be slightly smaller in  $d = 3$  (but this is far from certain); and  $z_{1C,CPU}$  seems to be significantly smaller in  $d = 4$ . Better measurements are definitely needed.

A second generalization, which works only in two dimensions, augments the SW algorithm by transformations to the dual lattice [47]. Preliminary data show the complete elimination of critical slowing-down: for the energy  $E$  — but *not* for other observables, such as  $(E - \langle E \rangle)^2$  — in the “ultra-scaling region”  $|\beta - \beta_c| \ll L^{-1/\nu}$ .

The idea of the multi-scale SW algorithm [51] is to carry out only a partial FKS algorithm, but then to apply this concept recursively in a multi-grid style. This is a very interesting idea, and it is plausible that the *W-cycle* version of this algorithm could have a dynamic critical exponent smaller than that of standard SW. But this remains to be established; the claims that  $z = 0$  are in my opinion unsupported by the *currently* available numerical evidence.

The SW algorithm can be generalized in a straightforward manner to Potts lattice gauge theories (more precisely, lattice gauge theories with a *finite abelian* gauge group  $G$  and Potts ( $\delta$ -function) action). The auxiliary variables  $\{n\}$  live now on *plaquettes*, and the SW update requires generating a random gauge field subject to the constraint of zero curvature on each occupied plaquette. Doing this efficiently (i.e. in time of order volume) seems

to be a difficult problem in computational algebraic topology (an almost nonexistent field [48]); it has been done in  $d = 3$  by a clever use of duality [49]. [I emphasize that “clusters” play no role in this algorithm: clusters (= connected components) are zeroth cohomology, whereas what is relevant here is first cohomology mod  $G$ .] Experiments on the three-dimensional  $Z_2$  gauge theory [49,50] yield a dynamic critical exponent  $z \approx 0.6 - 0.7$ .

The SW algorithm can easily be generalized to Ising models with both ferromagnetic and antiferromagnetic couplings, but this generalized algorithm does *not* work well [23]. The trouble is that the ferromagnetic and antiferromagnetic bonds work against each other in making a phase transition, but work together in making the SW bonds percolate; therefore, the SW bonds begin to percolate well above the critical temperature, and near criticality almost all the lattice belongs to a single huge cluster. (Flipping a huge cluster is equivalent to flipping its complement, which consists of many very small clusters.) Very recently, the Weizmann group has made significant advance toward handling frustrated Ising models [52]: their idea is to consider all the bonds in a plaquette as a single unit, and to apply a cleverly chosen FKSW transformation to this entity. They say that their method works well when *every* plaquette is frustrated, but not in cases of partial frustration. This may be an area of rapid progress in the next year or two.

Finally, it is worth indicating the general idea behind all algorithms of Swendsen-Wang type. Consider an arbitrary statistical-mechanical model with variables  $\{\varphi\}$ , and let  $W(\{\varphi\})$  be its Boltzmann weight. The idea is then to introduce auxiliary variables  $\{n\}$  according to

$$W(\{\varphi\}) = \sum_{\{n\}} W^{(n)}(\{\varphi\}). \quad (16)$$

Here the  $\{n\}$  are any kind of variables you like, discrete or continuous (in the latter case the sum would be an integral), and you are free to decompose  $W$  into partial Boltzmann weights  $W^{(n)}$  any way you please. Usually  $W = \prod_b W_b$  where the “bonds”  $b$  are sites, links or plaquettes, and correspondingly  $W^{(n)} = \prod_b W_{n_b}$ ; but this is not manda-

tory. One then simulates the joint model

$$W_{\text{joint}}(\{\varphi\}, \{n\}) = W^{(n)}(\{\varphi\}) \quad (17)$$

by any legal algorithm (usually by alternately applying the conditional distributions given  $\{\varphi\}$  and  $\{n\}$ ). Such an algorithm may or may not reduce the critical slowing-down; that depends on the *physics* behind the decomposition (16). But this formalism provides, in any event, an easy way to check the validity of proposed SW-type algorithms.

## 5. EMBEDDING ALGORITHMS

Thanks to Swendsen and Wang, we now have a fantastically good algorithm for simulating ferromagnetic Ising (and Potts) models. Can we extend our success to non-Potts models such as nonlinear  $\sigma$ -models and lattice gauge theories?<sup>††</sup>

The most promising methods at present seem to be the *embedding algorithms*: the idea is to “embed” Ising variables  $\{\varepsilon\}$  “inside” the original model, and then simulate the induced Ising model using the ordinary SW algorithm (or the single-cluster variant).

For one-component spins, this embedding is the obvious decomposition into magnitude and sign [54]. Let the Hamiltonian be

$$H(\varphi) = -\beta \sum_{\langle xy \rangle} \varphi_x \varphi_y + \sum_x V(\varphi_x), \quad (18)$$

where  $\beta \geq 0$  and  $V(\varphi) = V(-\varphi)$ . We write

$$\varphi_x = \varepsilon_x |\varphi_x|, \quad (19)$$

<sup>††</sup>More than one conference participant asked me: “When are you guys [and gals] going to do something *useful*, like giving us an algorithm for QCD with dynamical fermions?” To such narrow minds I offer the same counsel that conservatives have always offered the poor: patience. More precisely, I advise using (for now) the hybrid algorithm with a carefully tuned random trajectory length (this may achieve  $z \approx 1$  [26]), possibly combined with Fourier acceleration. Then wait a decade or so: I trust that by Lattice '00 we will hear either a decent numerical solution of QCD (using a few years CPU time on a teraflop parallel computer) or else a decent algorithm for QCD with dynamical fermions. (In truth, the situation is not quite so grim: there is a fair chance of progress in the near future on algorithms for computing propagators [6,7]; and this step consumes most of the CPU time in QCD computations.)



where  $\varepsilon_x \equiv \text{sgn}(\varphi_x) = \pm 1$ . For fixed values of the magnitudes  $\{|\varphi_x|\}$ , the conditional probability distribution of the  $\{\varepsilon\}$  is given by an Ising model with ferromagnetic (though space-dependent) couplings  $J_{xy} \equiv \beta_{|\varphi_x||\varphi_y|}$ . Therefore, the  $\{\varepsilon\}$  model can be updated efficiently by the Swendsen-Wang algorithm (or its single-cluster variant). Heat-bath or MGMCM sweeps must also be performed, in order to update the magnitudes.

Wolff's embedding algorithm [24,25] for  $O(N)$ -invariant spin models ( $N \geq 2$ ), independently invented by Hasenbusch [55], is equally simple. Let the Hamiltonian be

$$H(\sigma) = -\beta \sum_{\langle xy \rangle} \sigma_x \cdot \sigma_y + \sum_x V(|\sigma_x|), \quad (20)$$

with  $\beta \geq 0$ . Now fix a unit vector  $r \in \mathbb{R}^N$ , and write

$$\sigma_x = \sigma_x^\perp + \varepsilon_x |\sigma_x \cdot r| r, \quad (21)$$

where  $\sigma_x^\perp \equiv \sigma_x - (\sigma_x \cdot r)r$  and  $\sigma_x^\parallel \equiv (\sigma_x \cdot r)r$  are the components of  $\sigma_x$  perpendicular and parallel to  $r$ , and  $\varepsilon_x \equiv \text{sgn}(\sigma_x \cdot r) = \pm 1$ . (Flipping  $\varepsilon_x$  corresponds to reflecting  $\sigma_x$  in the hyperplane perpendicular to  $r$ .) Therefore, for fixed values of the  $\{\sigma^\perp\}$  and  $\{|\sigma \cdot r|\}$ , the probability distribution of the  $\{\varepsilon\}$  is given by an Ising model with ferromagnetic couplings  $J_{xy} \equiv \beta |\sigma_x \cdot r| |\sigma_y \cdot r|$ . The algorithm is then: Choose at random a unit vector  $r$ ; fix the  $\{\sigma^\perp\}$  and  $\{|\sigma \cdot r|\}$  at their current values, and update the  $\{\varepsilon\}$  by either SW or 1CSW. No other moves are required: the random choice of  $r$  suffices to make the algorithm ergodic.

A third example was provided recently by Evertz, Hasenbusch, Marcu, Pinn and Solomon [56]: it concerns solid-on-solid (SOS) models such as the discrete Gaussian model. An SOS model has integer-valued fields  $n_x$  living on lattice sites, and the Hamiltonian is of the form

$$H(n) = \sum_{\langle xy \rangle} V(|n_x - n_y|). \quad (22)$$

The embedding is as follows: Choose cleverly a "reflection level"  $M \in \mathbb{Z}$  or  $\mathbb{Z} + \frac{1}{2}$  (this is the subtle part), and write

$$n_x = M + \varepsilon_x |n_x - M|. \quad (23)$$

One then fixes the  $\{|n_x - M|\}$  and updates the  $\{\varepsilon\}$  by SW or 1CSW. This embedding is closely analogous to the Wolff embedding (21) for the XY model, if one identifies an SOS height with an XY angle.

A fourth example was provided by Ben-Av, Evertz, Marcu and Solomon [57]: it concerns the  $SU(2)$  lattice gauge theory at finite temperature, but only at  $N_c = 1$ !

The general idea behind all these algorithms is the following: "Foliate" the configuration space of the original model into "leaves" isomorphic to the configuration space of the "embedded" model. [In the above examples the embedded model is an Ising model, but the idea is much more general. For example, one might consider embeddings of XY spins in a higher  $\sigma$ -model,  $U(1)$  spins in an  $SU(N)$  gauge theory, etc.] One then moves around the current leaf, using any legitimate Monte Carlo algorithm for simulating the conditional probability distribution restricted to that leaf (i.e. the induced Hamiltonian for the embedded model). Of course, one must combine this with other moves, or with a different foliation, in order to make the algorithm ergodic. (This same structure arises in multi-grid Monte Carlo, where it is termed "partial resampling" [11,21,14].)

The performance of an embedding algorithm is determined by the combined effect of two completely distinct issues:

- i) How well the embedding captures the important large-scale collective modes of the original model.
- ii) How well some particular algorithm (e.g. standard SW or single-cluster SW) succeeds in updating the embedded model.

I wish to emphasize the importance of studying these questions, *separately*. If the physically relevant large-scale collective motions of the original model cannot be obtained by motions *within* a leaf, then the embedding algorithm will have severe critical slowing-down *no matter what* method is used to update the embedded variables. On the other hand, if the embedding algorithm with a particular choice of updating method for the embed-

ded variables shows severe critical slowing-down, this does *not* necessarily mean that the embedding works badly: the poor performance might be due to slow decorrelation in the inner updating subroutine, and could possibly be remedied by switching to a better algorithm for updating the embedded model. (This is particularly likely to occur if the induced Hamiltonian for the embedded model exhibits *frustration*.) It is crucial to distinguish these two issues, if we wish to obtain *physical insight*.

How can we disentangle these two effects? To study question (i), we can investigate the *idealized embedding algorithm* defined by *independent resampling* on each leaf. (In MGMC this is called the “idealized two-grid algorithm”.) To approximate this in practice, one makes  $N_{hit}$  hits of the best available method for simulating the embedded model, and extrapolates to  $N_{hit} = \infty$ . (A familiar analogue is approximating the single-site heat-bath algorithm by multi-hit Metropolis.) I emphasize that this is *not* claimed to be an efficient algorithm (usually  $N_{hit} = 1$  is optimal for fixed CPU time). Rather, it is a *test procedure* for gaining physical insight into the embedding; it is expensive but indispensable. To study question (ii), one can investigate the autocorrelation behavior of particular algorithms for the embedded model, using the induced Hamiltonians generated from a few “typical” configurations of the original model.

Let’s make these ideas concrete by looking at the Wolff algorithm for the  $N$ -vector model. At first thought it may seem strange (and somehow “unphysical”) to try to find Ising-like (i.e. discrete) variables in a model with a *continuous* symmetry group. However, upon reflection (pardon the pun) one sees what is going on [58]: if the spin configuration is slowly varying (e.g. a long-wavelength spin wave), then the induced Ising Hamiltonian tends to decouple along the surfaces where  $J_{xy}$  is small, hence where  $\sigma \cdot r \approx 0$ . The regions where  $\sigma \cdot r > 0$  and  $\sigma \cdot r < 0$  then get flipped *independently*, and this corresponds to a long-wavelength collective mode (Figure 1). So it is quite plausible that the *idealized* Wolff algorithm could have very small (or even zero) critical slowing-down in models where the important large-scale collective

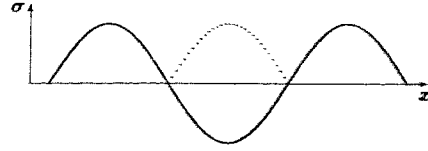


Figure 1: Action of the Wolff algorithm on a long-wavelength spin wave. For simplicity, both spin space ( $\sigma$ ) and physical space ( $x$ ) are depicted as one-dimensional.

modes are spin waves. (An additional argument [58] is needed to explain how the Wolff embedding deals with vortices in the two-dimensional  $XY$  model.) To see why the *practical* Wolff algorithm using SW or 1CSW updates also works well, it suffices to note that the induced Ising Hamiltonian is *ferromagnetic*, and that for such an Ising model SW and 1CSW work well.

Numerical tests of the Wolff algorithm confirm these predictions. For the two-dimensional models with  $N = 2, 3, 4$ , the data show  $z \lesssim 0.1$ , both in the idealized algorithm and in the practical algorithm with SW [58] or 1CSW [25] updates. For the three-dimensional  $XY$  model, a simulation using standard SW updates ( $N_{hit} = 1$ ) found  $z_{i,m,f} \approx 0.46$  [59], while one using single-cluster updates found  $z_{int,f,CPU} \approx 0.25$  [60]. But these latter exponents may well be due to critical slowing-down in the inner SW or 1CSW subroutine; a study of the *idealized* Wolff algorithm for this model would be very useful [61].

In view of the extraordinary success of the Wolff algorithm for spin models, it is tempting to try to extend it to lattice gauge theories with continuous gauge group [for example,  $U(1)$ ,  $SU(N)$  or  $SO(N)$ ]. Gauge theories differ from  $N$ -vector models in two ways:

- a) The field takes values in a *group* rather than a sphere. [ $U(1)$  and  $SU(2)$  are spheres, but higher Lie groups are not.]
- b) The field is a 1-form rather than a 0-form, i.e. it lives on *links* rather than sites. Correspond-

ingly, the energy is the curl of the field rather than its gradient, and it lives on plaquettes rather than links. As a result, the theory has a local gauge invariance rather than just a global symmetry.

The deep physical difference between gauge and spin models is, of course, item (b). The fact of gauge invariance, and the transverseness of physical excitations in a gauge theory, will impose severe constraints on the as-yet-unknown analogue of the embedding (21) [if indeed such an analogue exists].<sup>††</sup> At present I have little to say in this direction (though some insight might possibly be gleaned from the Swendsen-Wang algorithm for Potts lattice gauge theories [49,50]). Instead, Caracciolo, Edwards, Pelissetto and I [62] have addressed the less profound, but still highly nontrivial, problem (a). To do this, we ask whether the embedding (21) can be generalized to nonlinear  $\sigma$ -models taking values in manifolds other than spheres — such as  $SU(N)$  for  $N \geq 3$  — and, if so, what is the dynamic critical behavior of the corresponding idealized Wolff algorithm. Our approach is as follows: First we ask what are the fundamental properties of the embedding (21) that cause the Wolff algorithm to work so well. Then we ask whether embeddings having these properties exist also in other Riemannian manifolds  $M$ ; this is a question in differential geometry to which we are able to give a fairly complete answer. Finally, we perform a numerical study to test (in one case) whether our theoretical reasoning is correct. The conclusion of this analysis is quite surprising: roughly speaking, we find that a generalized Wolff algorithm can work well (i.e. have  $z \ll 2$ ) only if the manifold  $M$  is a Cartesian product of one or more spaces of constant positive curvature. This means that  $M$  is either a sphere, or the quotient of a sphere by a discrete group (for example, real projective space  $RP^{N-1}$ ), or a product of such spaces. If correct, this conclusion is quite disappointing, and lends renewed impetus to other classes of collective-mode algorithms such as multi-

grid Monte Carlo and Fourier acceleration.

#### ACKNOWLEDGMENTS

This research was supported in part by the U.S. National Science Foundation grant DMS-8911273, and by an Outstanding Junior Investigator Award from the U.S. Department of Energy (contract DE-FG02-90ER40581).

#### REFERENCES

1. A.D. Sokal, in *VIII<sup>th</sup> International Congress on Mathematical Physics*, ed. R. Sénéor and M. Mebkhout (World Scientific, Singapore, 1987).
2. A.D. Sokal, in *Computer Simulation Studies in Condensed Matter Physics: Recent Developments*, ed. D.P. Landau, K.K. Mon and H.-B. Schüttler (Springer-Verlag, Berlin-Heidelberg, 1988).
3. S.L. Adler, *Nucl. Phys. B (Proc. Suppl.)* **9**, 437 (1989).
4. U. Wolff, *Nucl. Phys. B (Proc. Suppl.)* **17**, 93 (1990).
5. D. Weingarten, *Nucl. Phys. B (Proc. Suppl.)* **9**, 447 (1989).
6. R.G. Edwards, J. Goodman and A.D. Sokal, *Nucl. Phys. B (Proc. Suppl.)* **9**, 521 (1989).
7. Talks presented at this conference by R. Brower, M. Hamzatz and A. Hulsebos.
8. Talk presented at this conference by A. Pelissetto, and references cited therein.
9. M.E. Agishtein and A.A. Migdal, *Int. J. Mod. Phys. C1*, 165 (1990).
10. J.M. Hammersley and D.C. Handscomb, *Monte Carlo Methods* (Methuen, London, 1964), Chapter 9.
11. 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).
12. N. Madras and A.D. Sokal, *J. Stat. Phys.* **50**, 109 (1988).
13. J. Goodman and N. Madras, Courant Institute preprint (1987).
14. J. Goodman and A.D. Sokal, *Phys. Rev. D* **40**, 2035 (1989).

<sup>††</sup>The same issue arises in devising multi-grid algorithms for gauge theories [14, Section V].

15. Ising models: G.F. Mazenko and O.T. Valls, *Phys. Rev.* **B24**, 1419 (1981); C. Kalle, *J. Phys.* **A17**, L801 (1984); J.K. Williams, *J. Phys.* **A18**, 49 (1985); R.B. Pearson, J.L. Richardson and D. Toussaint, *Phys. Rev.* **B31**, 4472 (1985); S. Wansleben and D.P. Landau, *J. Appl. Phys.* **61**, 3968 (1987); N. Ito, M. Taiji and M. Suzuki, *J. Phys. Soc. Japan* **56**, 4218 (1987); N. Ito, M. Taiji and M. Suzuki, *J. Physique (Suppl.)* **49**, C8-1397 (1988).
16. Potts models: For a summary, see S. Tang and D.P. Landau, *Phys. Rev.* **B36**, 567 (1987), Table I.
17. U. Wolff, *Phys. Lett.* **B228**, 379 (1989).
18. M.N. Barber, in *Phase Transitions and Critical Phenomena*, vol. 8, ed. C. Domb and J.L. Lebowitz (Academic Press, London, 1983); K. Binder, *J. Comput. Phys.* **59**, 1 (1985); V. Privman, ed., *Finite Size Scaling and Numerical Simulation of Statistical Systems* (World Scientific, Singapore, 1990).
19. K. Kremer and K. Binder, *Comput. Phys. Reports* **7**, 259 (1988).
20. G. Parisi, in *Progress in Gauge Field Theory*, ed. G. 't Hooft *et al.* (Plenum, New York, 1984); G.G. Batrouni, G.R. Katz, A.S. Kronfeld, G.P. Lepage, B. Svetitsky and K.G. Wilson, *Phys. Rev.* **D32**, 2736 (1985); J.B. Kogut, *Nucl. Phys.* **B275** [F517], 1 (1986); S. Duane, R. Kenway, B.J. Pendleton and D. Roweth, *Phys. Lett.* **B176**, 143 (1986); E. Dagotto and J.B. Kogut, *Phys. Rev. Lett.* **58**, 299 (1987) and *Nucl. Phys.* **B290** [F520], 451 (1987).
21. J. Goodman and A.D. Sokal, *Phys. Rev. Lett.* **56**, 1015 (1986).
22. R.G. Edwards, J. Goodman and A.D. Sokal, *Nucl. Phys. B* (to appear).
23. R.H. Swendsen and J.-S. Wang, *Phys. Rev. Lett.* **58**, 86 (1987).
24. U. Wolff, *Phys. Rev. Lett.* **62**, 361 (1989).
25. U. Wolff, *Nucl. Phys.* **B322**, 759 (1989); *Nucl. Phys.* **B334**, 581 (1990).
26. B. Pendleton, talk presented at this conference.
27. A. Hulsebos, J. Smit and J.C. Vink, Multi-grid simulation of the  $XY$  model, Amsterdam preprint ITFA-90-17 (August 1990).
28. S. Meyer, talk presented at this conference.
29. R.G. Edwards, S.J. Ferreira, J. Goodman and A.D. Sokal, in preparation.
30. Exact Langevin: P.J. Rossky, J.D. Doll and H.L. Friedman, *J. Chem. Phys.* **69**, 4628 (1978); R.T. Scalettar, D.J. Scalapino and R.L. Sugar, *Phys. Rev.* **B34**, 7911 (1986); M. Creutz, *Phys. Rev.* **D38**, 1228 (1988).
31. Exact hybrid: S. Duane, A.D. Kennedy, B.J. Pendleton and D. Roweth, *Phys. Lett.* **B195**, 216 (1987); M. Creutz, *Phys. Rev.* **D38**, 1228 (1988); H. Gausterer and S. Sanielevici, *Phys. Rev.* **D38**, 1220 (1988); R. Gupta, G.W. Kilcup and S.R. Sharpe, *Phys. Rev.* **D38**, 1278 (1988); A.D. Kennedy, *Nucl. Phys. B (Proc. Suppl.)* **9**, 457 (1989).
32. F.Y. Wu, *Rev. Mod. Phys.* **54**, 235 (1982); **55**, 315 (E) (1983).
33. E.M. Reingold, J. Niervergelt and N. Deo, *Combinatorial Algorithms: Theory and Practice* (Prentice-Hall, Englewood Cliffs, N.J., 1977), Chapter 8; A. Gibbons, *Algorithmic Graph Theory* (Cambridge University Press, 1985), Chapter 1; S. Even, *Graph Algorithms* (Computer Science Press, Potomac, Maryland, 1979), Chapter 3.
34. B.A. Galler and M.J. Fischer, *Commun. ACM* **7**, 301, 506 (1964); D.E. Knuth, *The Art of Computer Programming*, vol. 1, 2<sup>nd</sup> ed., (Addison-Wesley, Reading, Massachusetts, 1973), pp. 353-355, 360, 572; J. Hoshen and R. Kopelman, *Phys. Rev.* **B14**, 3438 (1976).
35. Y. Shiloach and U. Vishkin, *J. Algorithms* **3**, 57 (1982).
36. P.W. Kasteleyn and C.M. Fortuin, *J. Phys. Soc. Japan* **26** (Suppl.), 11 (1969); C.M. Fortuin and P.W. Kasteleyn, *Physica* **57**, 536 (1972); C.M. Fortuin, *Physica* **58**, 393 (1972); C.M. Fortuin, *Physica* **59**, 545 (1972).
37. C.-K. Hu, *Phys. Rev.* **B29**, 5103 (1984); T.A. Larsson, *J. Phys.* **A19**, 2383 (1986) and **A20**, 2239 (1987).
38. R.G. Edwards and A.D. Sokal, *Phys. Rev.* **D38**, 2009 (1988).
39. N. Ito, M. Taiji and M. Suzuki, *J. Phys. Soc. Japan* **56**, 4218 (1987).
40. D.W. Heermann and A.N. Burkitt, *Physica* **A162**, 210 (1990).
41. C.F. Baillie and P.D. Coddington, *Nucl. Phys. B (Proc. Suppl.)* **17**, 305 (1990).

42. X.-J. Li and A.D. Sokal, *Phys. Rev. Lett.* **63**, 827 (1989).
43. J.-S. Wang, *Physica A* **104**, 240 (1990).
44. W. Klein, T. Ray and P. Tamayo, *Phys. Rev. Lett.* **62**, 163 (1989).
45. T.S. Ray, P. Tamayo and W. Klein, *Phys. Rev. A* **39**, 5949 (1989).
46. P. Tamayo, R.C. Brower and W. Klein, *J. Stat. Phys.* **58**, 1083 (1990).
47. R.G. Edwards and A.D. Sokal, *Swendsen-Wang algorithm augmented by duality transformations for the two-dimensional Potts model, in preparation.*
48. J. Saludes i Closa, *Bull. Soc. Catalana Cièn.* **3**, 127 (1984); I.K. Svetlichnova, *Vestnik Mosk. Univ. Mat.* **39**, no. 3, 79 (1984) [= *Moscow Univ. Math. Bull.* **39**, no. 5, 100 (1984)].
49. R. Ben-Av, D. Kandel, E. Katznelson, P.G. Laursen and S. Solomon, *J. Stat. Phys.* **56**, 125 (1990).
50. R.C. Brower and S. Huang, *Phys. Rev.* **B41**, 708 (1990).
51. D. Kandel, E. Domany, D. Ron, A. Brandt and E. Loh, *Phys. Rev. Lett.* **60**, 1591 (1988); D. Kandel, E. Domany and A. Brandt, *Phys. Rev.* **B40**, 330 (1989).
52. D. Kandel, R. Ben-Av and E. Domany, *Phys. Rev. Lett.* **65**, 941 (1990).
53. F. Niedermayer, *Phys. Rev. Lett.* **61**, 2026 (1988).
54. R.C. Brower and P. Tamayo, *Phys. Rev. Lett.* **62**, 1087 (1989).
55. M. Hasenbusch, *Nucl. Phys.* **B333**, 581 (1990).
56. S. Solomon, talk presented at this conference; H.G. Evertz, M. Hasenbusch, M. Marcu, K. Pinn and S. Solomon, *Phys. Lett. B* (to appear).
57. H.G. Evertz, talk presented at this conference.
58. R.G. Edwards and A.D. Sokal, *Phys. Rev.* **D40**, 1374 (1989).
59. M. Hasenbusch and S. Meyer, *Phys. Lett.* **B241**, 238 (1990).
60. W. Janke, *Phys. Lett.* **A148**, 306 (1990).
61. S. Caracciolo, R.G. Edwards, A. Pefissetto and A.D. Sokal, work in progress.
62. R.G. Edwards, talk presented at this conference; S. Caracciolo, R.G. Edwards, A. Pefissetto and A.D. Sokal, in preparation.