Exact Sampling with Coupled Markov Chains and Applications to Statistical Mechanics *

James Gary Propp
propp@math.mit.edu

David Bruce Wilson
dbwilson@mit.edu

Department of Mathematics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

July 16, 1996

Abstract

For many applications it is useful to sample from a finite set of objects in accordance with some particular distribution. One approach is to run an ergodic (i.e., irreducible aperiodic) Markov chain whose stationary distribution is the desired distribution on this set; after the Markov chain has run for M steps, with M sufficiently large, the distribution governing the state of the chain approximates the desired distribution. Unfortunately it can be difficult to determine how large M needs to be. We describe a simple variant of this method that determines on its own when to stop, and that outputs samples in exact accordance with the desired distribution. The method uses couplings, which have also played a role in other sampling schemes; however, rather than running the coupled chains from the present into the future, one runs from a distant point in the past up until the present, where the distance into the past that one needs to go is determined during the running of the algorithm itself. If the state space has a partial order that is preserved under the moves of the Markov chain, then the coupling is often particularly efficient. Using our approach one can sample from the Gibbs distributions associated with various statistical mechanics models (including Ising, random-cluster, ice, and dimer) or choose uniformly at random from the elements of a finite distributive lattice.

1. Introduction

There are a number of reasons why one might want a procedure for generating a combinatorial object "randomly"; for instance, one might wish to determine statistical properties of the class as a whole, with a view towards designing statistical tests of significance for experiments that give rise to objects in that class, or one might wish to determine the generic properties of members of a large finite class of structures. One instance of the first sort of motivation is the work of Diaconis and Sturmfels on design of chi-squared tests for arrays of non-negative integers with fixed row- and column-sums [21]; instances of the second sort of motivation are seen in the work of the first author and others on domino tilings [35] [45] and in the vast literature on Monte Carlo simulations in physics (Sokal gives a survey [51]).

In many of these cases, it is possible to devise an ergodic (i.e., irreducible and aperiodic) Markov chain (e.g., the Metropolis-Hastings Markov chain) on the set of objects being studied, such that the steady-state distribution of the chain is precisely the distribution π that one wishes to sample from (this is sometimes called "Markov chain Monte Carlo"). Given such a Markov chain, one can

*During the conduct of the research that led to this article, the first author was supported by NSA grant MDA904-92-H-3060 and NSF grant DMS 9206374, and the second author was supported in part by an ONR-NDSEG fellowship.

start from any particular probability distribution ρ, even one that is concentrated on a single state (that is, a single object in the set), and approach the desired distribution arbitrarily closely, simply by running the chain for a sufficiently long time (say for M steps), obtaining a new probability distribution ρ^M. The residual *initialization bias* after one has run the chain for M steps is defined as the *total variation distance* $\|\rho^M - \pi\|$, i.e., as $\frac{1}{2}\sum_i |\rho^M(i) - \pi(i)|$, or equivalently as the maximum of $|\rho^M(E) - \pi(E)|$ over all measurable events E; for more background on this and other basic facts about Markov chains, see [18] or [5]. Even if one is willing to settle for a small amount of bias in the sample, there is a hidden difficulty in the Markov chain Monte Carlo method: there is no simple way to determine *a priori* how big M must be to achieve $\|\rho^M - \pi\| < \varepsilon$ for some specified $\varepsilon > 0$. Recently there has been much work at analyzing the rates of convergence of Markov chains [18] [34] [20] [43] [50] [19] [48] but this remains an arduous undertaking. Thus, an experimenter who is using such random walk methods faces the risk that the Markov chain has not yet had time to equilibrate, and that it is exhibiting metastable behavior rather than true equilibrium behavior.

In section 2 we present an alternative approach, applicable in a wide variety of contexts, in which many Monte Carlo algorithms (some of them already in use) can be run in such a manner as to remove all initialization bias. Our scheme in effect uses the Markov chain in order to get an estimate of its own mixing time (cf. the work of Besag and Clifford [10], who use a Markov chain to assess whether or not a given state was sampled from π, while requiring and obtaining no knowledge of the Markov chain's mixing time). Our main tools are *couplings*, and the main ideas that we apply to them are *simulation from the past* and *monotonicity*. A coupled Markov chain has states that are ordered pairs (or more generally k-tuples) of states of the original Markov chain, with the property that the dynamics on each separate component coincide with the original Markov dynamics. The versatility of the coupling method lies in the amount of freedom one has in choosing the statistics of the coupling (see [42] for background on couplings).

Our approach is to simulate the Markov chain by performing random moves until some predetermined amount of time has elapsed, in the hope that all the states will "coalesce" in that time; if they do, we can output the resulting coalescent state as our sample. If the states have not coalesced, then we restart the chain further back in time, prepending new random moves to the old ones. We show that if enough moves are prepended, eventually the states will coalesce, and the result will be an unbiased random sample (subsection 2.1).

When the number of states in the Markov chain is large, the preceding algorithm is not feasible exactly as described. However, suppose (as is often the case) that we can impose a partial ordering on the state-space, and that not only do we have a Markov chain that preserves the probability distribution that we are trying to sample from, but we also have a way of coupling this Markov chain with itself that respects the partial ordering of the state-space under time-evolution. Then this coupling enables us to ascertain that coalescence has occurred merely by verifying that coalescence has occurred for the histories whose initial states were the maximal and minimal elements of the state space (subsection 2.2). We call this the *monotone Monte Carlo* technique. Often there is a unique minimal element and a unique maximal element, so only two histories need to be simulated. (For a related approach, see [36].)

In the case of finite *spin-systems*, there are a finite number of *sites* and each state is a configuration that assigns one of two *spins* ("up" or "down") to every site, and so there is a natural partial ordering with unique maximal and minimal elements. When the distribution π on the state-space is *attractive* (a term we will define precisely in subsection 3.1), then the *heat bath algorithm* on the spin-system preserves the ordering (compare with section III.2 of [41]). In this situation, the idea of simulating from the past and the monotone Monte Carlo technique work together smoothly and lead to a procedure for exact sampling. One offshoot of this is that one can sample from the uniform distribution on the set of elements of any finite distributive lattice (see subsection 3.2); this is in fact a more useful class of applications than one might at first suppose, since many interesting combinatorial structures can be viewed as elements of a distributive lattice in a non-obvious way (see subsection 3.3 and [47]).

Section 4 contains our main applications, in which the desired distribution π is the Gibbs distribution for a statistical mechanics model. A virtue of our method is that its algorithmic implementation is merely a variant of some approximate sampling algorithms that are already in use, and typically involves very little extra computational overhead. A dramatic illustration of our approach is given in subsection 4.2, where we show (using a clever trick due to Fortuin and Kasteleyn) that one can sample from the Gibbs distribution (see [51] and below) on the set of states of a general ferromagnetic Ising model in any number of dimensions without being subject to the "critical slowing down" that besets more straightforward approaches. Another strength of our method is that it sometimes permits one to obtain "omnithermal samples" (see subsections 3.1 and 4.2) that in turn may allow one to get estimates of critical exponents (see below).

In section 5 we show how the expected running time of the algorithm can be bounded in terms of the mixing time of the chain (as measured in terms of total variation distance). Conversely, we show how data obtained by running the algorithm give estimates on the mixing time of the chain. Such estimates, being empirically derived, are subject to error, but one can make rigorously provable statements about the unconditioned likelihood of one's estimates being wrong by more than a specified amount. A noteworthy feature of our analysis is that it demonstrates that *any* coupling of the form we consider is in fact an efficient coupling (in the sense of being nearly optimal).

It has not escaped our notice that the method of coupling from the past can be applied to the problem (also considered in [7], [1], [44], and [3]) of randomly sampling from the steady state of a Markov chain whose transition-probabilities are unknown but whose transitions can be simulated or observed. This problem will be treated in two companion papers [61], [60].

2. General Theory

2.1. Coupling from the past

Suppose that we have an ergodic (irreducible and aperiodic) Markov chain with n states, numbered 1 through n, where the probability of going from state i to state j is $p_{i,j}$. Ergodicity implies that there is a unique stationary probability distribution π, with the property that if we start the Markov chain in some state and run the chain for a long time, the probability that it ends up in state i converges to $\pi(i)$. We have access to a randomized subroutine $Markov()$ which given some state i as input produces some state j as output, where the probability of observing $Markov(i) = j$ is equal to $p_{i,j}$; we will assume that the outcome of a call to $Markov()$ is independent of everything that precedes the call. This subroutine gives us a method for approximate sampling from the probability distribution π, in which our Markov-transition oracle $Markov()$ is iterated M times (with M large) and then the resulting state is output. For convenience, we assume that the start and finish of the simulation are designated as time $-M$ and time 0; of course this notion of time is internal to the simulation being performed, and has nothing to do with the external time-frame in which the computations are taking place. To start the chain, we use an arbitrarily chosen initial state i^*:

$i_{-M} \leftarrow i^*$ (start chain in state i^* at time $-M$)

for $t = -M$ to -1

  $i_{t+1} \leftarrow Markov(i_t)$

return i_0

We call this *fixed-time forward simulation.* Unfortunately, it can be difficult to determine what constitutes a large enough M relative to any specified fidelity criterion. That is to say, if ρ denotes the point-measure concentrated at state i^*, so that ρ^M denotes the probability measure governing i_0, then it can be hard to assess how big M needs to be so as to guarantee that $\|\rho^M - \pi\|$ is small.

We will describe a procedure for sampling with respect to π that does not require foreknowledge of how large the cutoff M needs to be. We start by describing an approximate sampling procedure whose output is governed by the same probability distribution ρ^M as M-step forward simulation, but which starts at time 0 and moves into the past; this procedure takes fewer steps than fixed-time forward simulation when M is large. Then, by removing the cutoff M — by effectively setting it equal to infinity — we will see that one can make the simulation output state i with probability exactly $\pi(i)$, and that the expected number of simulation steps will nonetheless be finite. (Issues of efficiency will be dealt with in section 5 of our article, in the case where the Markov chain is monotone; for an analysis of the general case, see our article [61], in which we improve on the naive approach just discussed.)

To run fixed-time simulation backwards, we start by running the chain from time -1 to time 0. Since the state of the chain at time -1 is determined by the history of the chain from time $-M$ to time -1, that state is unknown to us when we begin our backwards simulation; hence, we must run the chain from time -1 to time 0 not just once but n times, once for each of the n states of the chain that might occur at time -1. That is, we can define a map f_{-1} from the state space to itself, by putting $f_{-1}(i) = Markov(i)$ for $i = 1, \ldots, n$. Similarly, for all times t with $-M \leq t < -1$, we can define a random map f_t by putting $f_t(i) = Markov(i)$ (using separate calls to $Markov()$ for each time t); in fact, we can suppress the details of the construction of the f_t's and imagine that each successive f_t is obtained by calling a randomized subroutine *RandomMap*() whose *values* are actually *functions* from the state space to itself. The output of fixed-time simulation is given by $F_{-M}^0(i^*)$, where $F_{t_1}^{t_2}$ is defined as the composition $f_{t_2-1} \circ f_{t_2-2} \circ \cdots \circ f_{t_1+1} \circ f_{t_1}$.

If this were all there were to say about backward simulation, we would have incurred a substantial computational overhead (vis-a-vis forward simulation) for no good reason. Note, however, that under backward simulation there is no need to keep track of all the maps f_t individually; rather, one need only keep track of the compositions F_t^0, which can be updated via the rule $F_t^0 = F_{t+1}^0 \circ f_t$. More to the point is the observation that if the map F_t^0 ever becomes a constant map, with $F_t^0(i) = F_t^0(i')$ for all i, i', then this will remain true from that point onward (that is, for all earlier t's), and the value of $F_{-M}^0(i^*)$ must equal the common value of $F_t^0(i)$ $(1 \leq i \leq n)$; there is no need to go back to time $-M$ once the composed map F_t^0 has become a constant map. When the map F_t^0 is a constant map, we say coalescence occurs from time t to time 0, or more briefly that coalescence occurs from time t. Backwards simulation is the procedure of working backward until $-t$ is sufficiently large that F_t^0 is a constant map, and then returning the unique value in the range of this constant map; if M was chosen to be large, then values of t that occur during the backwards simulation will almost certainly have magnitude much smaller than M.

We shall see below that as t goes to $-\infty$, the probability that the map F_t^0 is a constant map increases to 1. Let us suppose that the $p_{i,j}$'s are such that this typically happens with $t \approx -1000$. As a consequence of this, backwards simulation with M equal to one million and backwards simulation with M equal to one billion, which begin in exactly the same way, nearly always turn out to involve the exact same simulation steps (if one uses the same random numbers); the only difference between them is that in the unlikely event that $F_{-1,000,000}^0$ is not a constant map, the former algorithm returns the sample $F_{-1,000,000}^0(i^*)$ while the latter does more simulations and returns the sample $F_{-1,000,000,000}^0(i^*)$.

We now see that by removing the cut-off M, and running the backwards simulation into the past until F_t^0 is constant, we are achieving an output-distribution that is equal to the limit, as

M goes to infinity, of the output-distributions that govern fixed-time forward simulation for M steps. However, this limit is equal to π. Hence backwards simulation, with no cut-off, gives a sample whose distribution is governed by the steady-state distribution of the Markov chain. This algorithm may be stated as follows:

$t \leftarrow 0$

$F_t^0 \leftarrow$ the identity map

`repeat`

$t \leftarrow t-1$

$f_t \leftarrow RandomMap()$

$F_t^0 \leftarrow F_{t+1}^0 \circ f_t$

`until` $F_t^0(\cdot)$ is constant

`return` the unique value in the range of $F_t^0(\cdot)$

We remark that the procedure above can be run with $O(n)$ memory, where n is the number of states. The number of calls to $Markov()$ may also be reduced, and in a companion paper [61] we will show that the procedure can be modified so that its expected running time is bounded by a fixed multiple of the so-called *cover time* of the Markov chain (for a definition see [44]).

Theorem 1 *With probability 1 the coupling-from-the-past protocol returns a value, and this value is distributed according to the stationary distribution of the Markov chain.*

Proof: Since the chain is ergodic, there is an L such that for all states i and j, there is a positive chance of going from i to j in L steps. Hence for each t, $F_{t-L}^t(\cdot)$ has a positive chance of being constant. Since each of the maps $F_{-L}^0(\cdot), F_{-2L}^{-L}(\cdot), \ldots$ has some positive probability $\varepsilon > 0$ of being constant, and since these events are independent, it will happen with probability 1 that one of these maps is constant, in which case F_{-M}^0 is constant for all sufficiently large M. When the algorithm reaches back M steps into the past, it will terminate and return a value that we will call $\overline{F}_{-\infty}^0$. Note that $\overline{F}_{-\infty}^0$ is obtained from $\overline{F}_{-\infty}^{-1}$ by running the Markov chain one step, and that $\overline{F}_{-\infty}^0$ and $\overline{F}_{-\infty}^{-1}$ have the same probability distribution. Together these last two assertions imply that the output $\overline{F}_{-\infty}^0$ is distributed according to the unique stationary distribution π. $\square$

In essence, to pick a random element with respect to the stationary distribution, we run the Markov chain from the indefinite past until the present, where the distance into the past we have to look is determined dynamically, and more particularly, is determined by how long it takes for n runs of the Markov chain (starting in each of the n possible states, at increasingly remote earlier times) to coalesce.

In this set-up, the idea of simulating from the past up to the present is crucial; indeed, if we were to change the procedure and run the chain from time 0 into the future, finding the smallest M such that the value of $F_0^M(x)$ is independent of x and then outputting that value, we would obtain biased samples. To see this, imagine a Markov chain in which some states have a unique predecessor; it is easy to see such states can never occur at the exact instant when all n histories coalesce.

It is sometimes desirable to view the process as an iterative one, in which one successively starts up n copies of the chain at times -1, -2, etc., until one has gone sufficiently far back in the past to allow the different histories to coalesce by time 0. However, when one adopts this point of view (and we will want to do this in the next subsection), it is important to bear in mind that the random bits that one uses in going from time t to time $t+1$ must be the same for the many sweeps one might make through this time-step. If one ignores this requirement, then there will in general be bias in the samples that one generates. The curious reader may verify this by considering the Markov chain whose states are 0, 1, and 2, and in which transitions are implemented using a fair coin, by the rule that one moves from state i to state $\min(i+1, 2)$ if the coin comes up heads and to state $\max(i-1, 0)$ otherwise. It is simple to check that if one runs an incorrect version of our scheme in which entirely new random bits are used every time the chain gets restarted further into the past, the samples one gets will be biased in favor of the extreme states 0 and 2.

We now address the issue of independence of successive calls to $Markov()$ (implicit in our calls to the procedure $RandomMap()$). Independence was assumed in the proof of Theorem 1 in two places: first, in the proof that the procedure eventually terminates, and second, in the proof that the distribution governing the output of the procedure is the steady-state distribution of the chain. The use of independence is certainly inessential in the former setting, since it is clear that there are better ways to couple Markov chains than independently, if one's goal is to achieve rapid coalescence. The second use of independence can also be relaxed; what matters is that the random decisions made in updating the coupled chain from time t to time t are independent of the state of the chain at time t. This will be guaranteed if the random decisions made in updating from time t to time $t+1$ are independent of the random decisions made at all other times. Dependencies among the decisions made at a fixed time are perfectly legitimate, and do not interfere with this part of the proof. To treat possible dependencies, we adopt the point of view that $f_t(i)$, rather than being given by $Markov(i)$, is given by $\phi(i, U_t)$ where $\phi(\cdot,\cdot)$ is a deterministic function and U_t is a random variable associated with time t. We assume that the random variables $\ldots, U_{-2}, U_{-1}$ are i.i.d., and we take the point of view that the random process given by the U_t's is the source of all the randomness used by our algorithm, since the other random variables are deterministic functions of the U_t's. Note that this framework includes the case of full independence discussed earlier, since for example one could let the U_t's be i.i.d. variables taking their values in the n-cube $[0,1]^n$ with uniform distribution, and let $\phi(i,u)$ be the smallest j such that $p_{i,1} + p_{i,2} + \cdots + p_{i,j}$ exceeds the ith component of the vector u.

Theorem 2 *Let $\ldots, U_{-3}, U_{-2}, U_{-1}$ be i.i.d. random variables and $\phi(\cdot,\cdot)$ be a deterministic function with the property that for all i, $\mathrm{Prob}[\phi(i, U_{-1}) = j] = p_{i,j}$. Define $f_t(i) = \phi(i, U_t)$ and $F_t^0 = f_{-1} \circ f_{-2} \circ \cdots \circ f_t$. Assume that with probability 1, there exists t for which the map F_t^0 is constant, with a constant value that we may denote by $\phi(\ldots, U_{-2}, U_{-1})$. Then the random variable $\phi(\ldots, U_{-2}, U_{-1})$, which is defined with probability 1, has distribution governed by π.*

Rather than give a proof of the preceding result, we proceed to state and prove a more general result, whose extra generality, although not significant mathematically, makes it much closer to procedures that are useful in practice. The point of view here is that one might have several different Markovian update-rules on a state-space, and one might cycle among them. As long as each one of them preserves the distribution π, then the same claim holds.

Theorem 3 *Let $\ldots, U_{-3}, U_{-2}, U_{-1}$ be i.i.d. random variables and let $\phi_t(\cdot,\cdot)$ $(t < 0)$ be a sequence of deterministic functions with the property that for all t and j,*

$$\sum_i \pi(i)\mathrm{Prob}[\phi_t(i, U_t) = j] = \pi(j).$$

Define $f_t(i) = \phi_t(i, U_t)$ and $F_t^0 = f_{-1} \circ f_{-2} \circ \cdots \circ f_t$. Assume that with probability 1, there exists t for which the map F_t^0 is constant, with a constant value that we may denote by $\phi(\ldots, U_{-2}, U_{-1})$. Then *the random variable $\phi(\ldots, U_{-2}, U_{-1})$, which is defined with probability 1, has distribution governed by π.*

Proof: Let X be a random variable on the state-space of the Markov chain governed by the steady-state distribution π, and for all $t \leq 0$ let Y_t be the random variable $F_t^0(X)$. Each Y_t has distribution π, and the sequence $Y_{-1}, Y_{-2}, Y_{-3}, \ldots$ converges almost surely to some state $Y_{-\infty}$, which must also have distribution π. Put $\phi(\ldots, U_{-2}, U_{-1}) = Y_{-\infty}$. $\square$

We conclude this section by discussing two related ideas that appear in the published literature, and that were brought to our attention during the writing of this article.

The first is the method for generating random spanning trees of finite graphs due to Broder [15] and Aldous [4] in conversations with Persi Diaconis. In this method one does a primary random walk on the graph and "shadows" it with a secondary random walk on the set of spanning trees. The secondary walk is coalescent in the sense that if the primary walk visits all vertices of the graph, then the final state of the secondary walk is independent of its initial state. Thus we can apply coupling-from-the-past to get a random spanning tree, while following only a single history rather than the usual two histories in the monotone version of the method. The Broder-Aldous algorithm uses the fact that the simple random walk on a graph is reversible, and terminates in finite time with a random tree. Using the coupling-from-the-past approach rather than time-reversal, we can generalize this algorithm to yield random trees in a directed and weighted graph. Details will be given in [61]. (It is worth mentioning that for purposes of generating random trees, one can now do better than the Broder-Aldous algorithm; see [60].)

Time reversal also shows up in the theory of coalescent duality for interacting particle systems. A standard example of this duality is the relationship between the coalescing random walk model and the voter model generalized to weighted directed graphs (see [32], [27], [41]). Given a continuous-time Markov chain on n states, in which transitions from state i to state j occur at rate $p_{i,j}$, one defines a "coalescing random walk" by placing a particle on each state and decreeing that particles must move according to the Markov chain statistics; particles must move independently unless they collide, at which point they must stick together. The original Markov chain also determines a voter model, in which each of n voters starts by wanting to be be the leader, but in which voter i decides to adopt the current choice of voter j at rate $p_{i,j}$, until all voters eventually come into agreement on their choice of a leader. These two models are dual to one another, in the sense that each can be obtained from the other by simply reversing the direction of time (see [5]). As a corollary of the work above, the probability that voter i ends up as the leader is just the steady-state probability of state i in the original Markov chain.

2.2. Monotone Monte Carlo

Suppose now that the (possibly huge) state space S of our Markov chain admits a natural partial ordering $\leq$, and that our update rule ϕ has the property that $x \leq y$ implies $\phi(x, U_0) \leq \phi(y, U_0)$ almost surely with respect to U_0. Then we say that our Markov chain gives a *monotone Monte Carlo algorithm* for approximating π. We will suppose henceforth that S has elements $\hat{0}$, $\hat{1}$ with $\hat{0} \leq x \leq \hat{1}$ for all $x \in S$.

Define $\Phi_{t_1}^{t_2}(x, u) = \phi_{t_2-1}(\phi_{t_2-2}(\ldots(\phi_{t_1}(x, u_{t_1}), u_{t_1+1}), \ldots, u_{t_2-2}), u_{t_2-1})$, where u is short for $(\ldots, u_{-1}, u_0)$. If $u_{-T}, u_{-T+1}, \ldots, u_{-2}, u_{-1}$ have the property that $\Phi_{-T}^0(\hat{0}, u) = \Phi_{-T}^0(\hat{1}, u)$, then the monotonicity property assures us that $\Phi_{-T}^0(x, u)$ takes on their common value for all $x \in S$. This frees us from the need to consider trajectories starting in all $|S|$ possible states; two states will suffice. Indeed, the smallest T for which $\Phi_{-T}^0(\cdot, u)$ is constant is equal to the smallest T for which $\Phi_{-T}^0(\hat{0}, u) = \Phi_{-T}^0(\hat{1}, u)$.

Let T_* denote this smallest value of T. It would be possible to determine T_* exactly by a bisection technique, but this would be a waste of time: an overestimate for T_* is as good as the correct value, for the purpose of obtaining an unbiased sample. Hence, we successively try $T = 1, 2, 4, 8, \ldots$ until we find a T of the form 2^k for which $\Phi^0_{-T}(\hat{0}, u) = \Phi^0_{-T}(\hat{1}, u)$. The number of simulation-steps involved is $2(1 + 2 + 4 + \cdots + 2^k) < 2^{k+2}$, where the factor of 2 in front comes from the fact that we are simulating two copies of the chain (one from $\hat{0}$ and one from $\hat{1}$). However, this is close to optimal, since T_* must exceed 2^{k-1} (otherwise we would not have needed to go on to try $T = 2^k$); that is, the number of simulation steps required merely to *verify* that $\Phi^0_{-T_*}(\hat{0}, u) = \Phi^0_{-T_*}(\hat{1}, u)$ is greater than $2 \cdot 2^{k-1} = 2^k$. Hence our double-until-you-overshoot procedure comes within a factor of 4 of what could be achieved by a clairvoyant version of the algorithm in which one avoids overshoot.

Here is the pseudocode for our procedure.

$T \leftarrow 1$

`repeat`

  $upper \leftarrow \hat{1}$

  $lower \leftarrow \hat{0}$

  `for` $t = -T$ `to` -1

    $upper \leftarrow \phi_t(upper, u_t)$

    $lower \leftarrow \phi_t(lower, u_t)$

  $T \leftarrow 2T$

`until` $upper = lower$

`return` $upper$

Implicit in this pseudocode is the random generation of the u_t's. Note that when the random mapping $\phi_t(\cdot, u_t)$ is used in one iteration of the repeat loop, for any particular value of t, it is essential that the same mapping be used in all subsequent iterations of the loop. We may accomplish this by storing the u_t's; alternatively, if (as is typically the case) our u_t's are given by some pseudo-random number generator, we may simply suitably reset the random number generator to some specified seed $seed(i)$ each time t equals -2^i.

In the context of monotone Monte Carlo, a hybrid between fixed-time forward simulation and coupling-from-the-past is a kind of adaptive forward simulation, in which the monotone coupling allows one to check that the fixed time M that has been adopted is indeed large enough to guarantee mixing. This approach was foreshadowed by the work of Valen Johnson [36], and it has been used by Kim, Shor, and Winkler [38] in their work on random independent subsets of certain graphs; see subsection 3.3. Indeed, if one chooses M large enough, the bias in one's sample can be made as small as one wishes. However, it is worth pointing out that for a small extra price in computation time (or even perhaps a saving in computation time, if the M one chose was a very conservative estimate of the mixing time), one can use coupling-from-the-past to eliminate the initialization bias entirely.

In what sense does our algorithm solve the dilemma of the Monte Carlo simulator who is not sure how to balance his need to get a reasonably large number of samples and his need to get unbiased samples? We will see below that the expected run time of the algorithm is not much larger than the mixing time of the Markov chain (which makes it fairly close to optimal), and that the tail distribution of the run time decays exponentially quickly. If one is willing to wait for the algorithm to return an answer, then the result will be an unbiased sample. More generally, if one wants to generate several samples, then as long as one completes every run that gets started, all of the samples will be unbiased as well as independent of one another. Therefore, we consider our approach to be a practical solution to the Monte Carlo simulator's problem of not knowing how long to run the Markov chain.

We emphasize that the experimenter must not simply interrupt the current run of the procedure and discard its results, retaining only those samples obtained during earlier runs; the experimenter must either allow the run to terminate or else regard the final sample as indeterminate (or only partly determined) — or resign himself to contaminating his set of samples with bias. To reduce the likelihood of ending up in this dilemma, the experimenter can use the techniques of section 5.1 to estimate in advance the average number of Markov chain steps that will be required for each sample.

Recently Jim Fill has found an *interruptible* exact sampling protocol. Such a protocol, when allowed to run to completion, returns a sample distributed according to π, just as coupling-from-the-past does. Additionally, if an impatient user interrupts some runs, then rather than regarding the samples as indeterminate, the experimenter can actually throw them out without introducing bias. This is because the run time of an interruptible exact sampling procedure is independent of the sample returned, and therefore independent of whether or not the user became impatient and aborted the run.

As of yet there remain a number of practical issues that need to be resolved before a truly interruptible exact sampling program can be written. We expect that Fill and others will discuss these issues more thoroughly in future articles.

3. Spin Systems and Distributive Lattices

3.1. Attractive spin systems

Define a *spin system* on a vertex set V as the set of all ways of assigning a spin $\sigma(i)$ ("up" or "down") to each of the vertices $i \in V$, together with a probability distribution π on the set of such assignments $\sigma(\cdot)$. We order the set of configurations by putting $\sigma \geq \tau$ iff $\sigma(i) \geq \tau(i)$ for all $i \in V$ (where $\uparrow > \downarrow$), and we say that π is *attractive* if the conditional probability of the event $\sigma(i) = \uparrow$ is a monotone increasing function of the values of $\sigma(j)$ for $j \neq i$. (In the case of the Ising model, this corresponds to the ferromagnetic situation.)

More formally, given $i \in V$ and configurations σ, τ with $\sigma(j) \leq \tau(j)$ for all $j \neq i$, define configurations $\sigma_\uparrow$, $\sigma_\downarrow$, $\tau_\uparrow$, and $\tau_\downarrow$ by putting $\sigma_\uparrow(i) = \uparrow$, $\sigma_\uparrow(j) = \sigma(j)$ for all $j \neq i$, and so on. We say π is monotone iff $\pi(\sigma_\downarrow)/\pi(\sigma_\uparrow) \geq \pi(\tau_\downarrow)/\pi(\tau_\uparrow)$, or rather, if $\pi(\sigma_\downarrow)\pi(\tau_\uparrow) \geq \pi(\sigma_\uparrow)\pi(\tau_\downarrow)$, for all $\sigma \leq \tau$ and all $i \in V$.

A *heat bath algorithm* on a spin-system is a procedure whereby one cycles through the vertices i (using any mixture of randomness and determinacy that guarantees that each i almost surely gets chosen infinitely often) and updates the value at site i in accordance with the conditional probability for π. One may concretely realize this update rule by putting

$$f_t(\sigma, u_t) = \begin{cases} \sigma_\downarrow & \text{if } u_t < \pi(\sigma_\downarrow)/(\pi(\sigma_\downarrow) + \pi(\sigma_\uparrow)) \\ \sigma_\uparrow & \text{if } u_t \geq \pi(\sigma_\downarrow)/(\pi(\sigma_\downarrow) + \pi(\sigma_\uparrow)) \end{cases}$$

where t is the time, u_t is some random variable distributed uniformly in $[0, 1]$ (with all the u_t's independent of each other), and σ is some configuration of the system. If π is attractive, then this realization of π gives rise to a coupling-scheme that preserves the ordering (just use the same u_t's in all copies of the chain). To see why this is true, notice that if $\sigma < \tau$, then the u_t-threshold for σ is higher than for τ, so that the event $f_t(\sigma, u_t)(i) = \uparrow$, $f_t(\tau, u_t)(i) = \downarrow$ is impossible.

We now may conclude:

Theorem 4 *If one runs the heat bath for an attractive spin-system under the coupling-from-the-past protocol, one will generate states of the system that are exactly governed by the target distribution π.*

In certain cases this method is slow (for instance, if one is sampling from the Gibbs distribution for the Ising model below at the critical temperature), but in practice we find that it works fairly quickly for many attractive spin-systems of interest. In any case, section 5 will show that in a certain sense the coupling-from-the-past version of the heat bath is no slower than heat bath.

We point out that, like the standard heat bath algorithm, ours can be accelerated if one updates many sites in parallel, provided that these updates are independent of one another. For instance, in the case of the Ising model on a square lattice, one can color the sites so that black sites have only white neighbors and vice versa, and it then becomes possible to alternate between updating all the white sites and updating all the black sites. In effect, one is decomposing the configuration σ as a pair of configurations σ_{white} and σ_{black}, and one alternates between randomizing one of them in accordance with the conditional distribution determined by the other one.

We also point out that in many cases, one is studying a one-parameter family of spin systems in which some quantity such as temperature or the strength of an external field is varying. If some parameter (say the temperature T) affects the ratio $\pi(\sigma_\uparrow) : \pi(\sigma_\downarrow)$ in a monotone (say increasing) way, then it is possible to make an "omnithermal" heat bath Markov chain, one that in effect generates simultaneous random samples for *all* values of the temperature. An omnithermal state σ assigns to each vertex i the set $c(i)$ of temperatures for which site i is spin-down. The sets $c(i)$ are "monotone" in the sense that if a temperature is in $c(i)$, so is every lower temperature; that is, when the temperature is raised, spin-down sites may become spin-up, but not vice versa. Given a site i of configuration σ and a random number u between 0 and 1, the heat bath update rule updates $c(i)$ to be the set of T for which $u < \pi_T(\sigma_{T,\downarrow})/(\pi_T(\sigma_{T,\downarrow}) + \pi_T(\sigma_{T,\uparrow}))$, where σ_T denotes σ at temperature T. For each T this update rule is just the ordinary heat bath, and monotonicity ensures that the new set $c(i)$ is monotone in the aforementioned sense. The omnithermal heat bath Markov chain is monotone with a maximum and minimum state, so monotone coupling-from-the-past may be applied.

In the case where all the spin sites are independent of one another, the trick of simultaneously sampling for all values of a parameter has been used for some time in the theory of random graphs [6, chapter 10] and percolation [28]. Holley [31] used an omnithermal Markov chain with two temperatures to give an alternate proof of the FKG inequality [24] governing attractive spin systems. More recently Grimmett has given a monotone omnithermal Markov chain for the bond-correlated percolation model [29] (another name for the random cluster model described in subsection 4.2) and used it to derive a number of properties about these systems. Interestingly, Grimmett's Markov chain is different from the omnithermal heat bath chain. See subsection 4.2 for further discussion of omnithermal sampling and its uses.

3.2. Order ideals and antichains

Let P be a finite partially ordered set, and call a subset I of P an *order ideal* if for all $x \in I$ and $y \leq x$, we have $y \in I$. The set of order ideals of P is denoted by $J(P)$; it is a distributive lattice under the operations of union and intersection, and it is a standard theorem (see [52]) that every finite distributive lattice is of the form $J(P)$ for some finite partially ordered set P.

To turn $J(P)$ into a spin-system, we let V be the set of elements of P, and we associate the order ideal I with the spin-configuration σ in which $\sigma(i)$ is $\uparrow$ or $\downarrow$ according to whether $i \in I$ or $i \notin I$. Let us give each spin-configuration that arises in this fashion equal probability $\pi(\sigma) = 1/|J(P)|$, and give the rest probability 0. Then it is easy to check that the ratio $\pi(\sigma_\uparrow)/\pi(\sigma_\downarrow)$ is 1/0, 0/1, or

1/1, according to whether $\sigma \setminus \{i\}$ is not an order ideal, $\sigma \cup \{i\}$ is not an order ideal, or both sets are order ideals. Indeed, it is not much harder to see that π is attractive, so that by running the heat bath algorithm in a coupling-from-the-past framework, we can generate a uniform random element of $J(P)$.

We remind the reader that there is a one-to-one correspondence between order ideals of P and antichains of P (sets of pairwise-incomparable elements of P); specifically, for every order ideal I of P, the set of maximal elements of I forms an antichain, and every antichain determines a unique order ideal I. Hence, sampling uniformly from the elements of a finite distributive lattice is equivalent to sampling uniformly from the set of order ideals of a general finite poset, which is in turn equivalent to sampling uniformly from the set of antichains of a general finite poset.

We also point out that the heat bath procedure can often be done in parallel for many $i \in P$ at once. Suppose we color the elements of P so that no element covers another element of the same color. (If P is graded, then its Hasse diagram is bipartite and two colors suffice.) We use these vertex-colors to assign colors to the edges of the Hasse diagram G of $J(P)$. For definiteness, let us focus on one color, called "red". If I is an order ideal, and K is the set of all red elements of P whose adjunction to or removal from I yields an order ideal, then a "red move" is the operation of replacing I by the union of $I \setminus K$ with a random subset of K. If one alternates red moves with blue moves and so on, then one will converge to the uniform distribution on $J(P)$.

Cautionary note: While in many cases of interest this procedure will quickly find a random order ideal of a poset, it is not difficult to construct examples where the run time is very large. For instance, let P be the poset of $2n$ elements numbered $1, \ldots, 2n$, such that $x < y$ if and only if $x \leq n < y$ in the standard order on the integers. Then the Hasse diagram G of $J(P)$ will be two hypercubes joined at a single vertex, and the time for the upper and lower order ideals to coalesce will be exponential in n. Note however, that because of the bottleneck in this graph, the (uncoupled) random walk on this graph also takes exponential time to get close to random.

3.3. Combinatorial applications

In this section, we describe a few examples of combinatorial objects that can be sampled by means of the techniques developed in this section. In the first two examples, there are other methods that can be applied, but it is still interesting to see how versatile our basic approach is.

Lattice paths. First, consider the set of all lattice-paths of length $a+b$ from the point $(a, 0)$ to the point $(0, b)$. There are $\binom{a+b}{a}$ such paths, and there are a number of elementary techniques that one can use in order to generate a path at random. However, let us define the number of *inversions* in such a path as the number of times that an upward step is followed (not necessarily immediately) by a leftward step, so that the path from $(a, 0)$ to (a, b) to $(0, b)$ has ab inversions, and let us decree that each lattice-path should have probability proportional to q to the power of the number of inversions, for some $q \geq 0$. It so happens in this case that one can work out exactly what the constant of proportionality is, because one can sum $q^{\#\text{ of inversions}}$ over all lattice-paths with two fixed endpoints (these are the coefficients of the so-called "Gaussian binomial coefficients" [52]), and as a result of this there exists an efficient bounded-time procedure for generating a random q-weighted lattice-path. However, one also has the option of making use of coupling-from-the-past, as we now explain.

If we order the set of the unit squares inside the rectangle with corners $(0, 0), (a, 0), (0, b), (a, b)$ by decreeing that the square with lower-left corner (i', j') is less than or equal to the square with lower-left corner (i, j) if and only if $i' \leq i$ and $j' \leq j$, then we see that the unit squares that lie below and to the left of a lattice-path that joins $(a, 0)$ and $(0, b)$ form an order ideal, and that there is indeed a one-to-one-correspondence between the lattice-paths and the order ideals. Moreover, the number of inversions in a lattice path corresponds to the cardinality of the order ideal, so the order ideal I has probability proportional to $q^{|I|}$. It is not hard to show for any finite distributive lattice, a probability distribution of this form is always attractive. Therefore, the method applies. When $q = 1$, roughly $n^3 \log n$ steps on average are needed in order to generate an unbiased sample, where $a \approx b \approx n$; details will appear in [59].

Permutations. For certain statistical applications, it is useful to sample permutations $\pi \in S_n$ on n items such that the probability of π is proportional to $q^{\mathrm{inv}(\pi)}$; this is sometimes called "Mallows' phi model through Kendall's tau". For background see [37] and [18] and the references contained therein. Recall that $\mathrm{inv}(\pi)$ denotes the number of inversions of π, that is, the number of pairs (i, j) such that $i < j$ and $\pi(i) > \pi(j)$. We represent the permutation π by the n-tuple $[\pi(1), \ldots, \pi(n)]$.

Consider the following Markov chain whose steady-state distribution is the aforementioned distribution. Pick a random pair of adjacent items. With probability $1/(q + 1)$ put the two in ascending order, i.e. "sort them", and with probability $q/(q+1)$ put them in descending order, i.e. "un-sort" them. It is clear that this Markov chain is ergodic and preserves the desired probability distribution.

Define a partial order on S_n by $\pi < \sigma$ if and only if π can be obtained from σ by sorting adjacent elements (this is called the weak Bruhat order [13]). The bottom element is the identity permutation $\hat{0} : i \mapsto i$ and the top element is the totally reversing permutation $\hat{1} : i \mapsto n + 1 - i$. The above Markov chain is a random walk on the Hasse diagram of this partial order. (See [52] for background on partially ordered sets.) This Markov chain, coupled with itself in the obvious way, does not preserve the partial order, even on S_3. However, it is still true that when the top and bottom states coalesce, all states have coalesced, as we will show below. In symbols, the claim is that $F_{t_1}^{t_2}(\hat{0}) = F_{t_1}^{t_2}(\hat{1})$ implies that $F_{t_1}^{t_2}(\cdot)$ is constant. Therefore the technique of coupling from the past can be applied to this Markov chain.

The reason that the top and bottom states determine whether or not all states get mapped to the same place is that suitable projections of the Markov chain *are* monotone. Given a permutation π, let $\theta_k(\pi)$ be an associated threshold function. That is, $\theta_k(\pi)$ is a sequence of 0's and 1's, with a 1 at location i if and only $\pi(i) > k$. Just as the Markov chain sorts or un-sorts adjacent sites in a permutation, it sorts or un-sorts adjacent sites in the 0-1 sequence. Indeed, these sequences of 0's and 1's correspond to lattice-paths of the kind considered in the first example.

A sequence s_1 of 0's and 1's *dominates* another such sequence s_2 if the partial sums of s_1 are at least as large as the partial sums of s_2, i.e., $\sum_{i=1}^{I} s_1(i) \geq \sum_{i=1}^{I} s_2(i)$ for all $0 \leq I \leq n$. The Markov chain preserves dominance in sequences of 0's and 1's. This may be checked by case-analysis. For each k, we have that if $\theta_k(F_{t_1}^{t_2}(\hat{0})) = \theta_k(F_{t_1}^{t_2}(\hat{1}))$, then $\theta_k(F_{t_1}^{t_2}(\cdot))$ is constant. But note that if two permutations differ, then they must differ in some threshold function. Hence these threshold functions determine the permutation, and $F_{t_1}^{t_2}(\cdot)$ must be constant if it maps $\hat{0}$ and $\hat{1}$ to the same place.

Recently Felsner and Wernisch reported having generalized this monotone Markov chain to sublattices of the weak Bruhat lattice. Given two permutations π_1 and π_2 with $\pi_1 \leq \pi_2$, the random walk is restricted to those permutations σ such that $\pi_1 \leq \sigma \leq \pi_2$. This Markov chain can be used to sample random linear extensions of a two-dimensional partially ordered set [22].

Independent sets. A natural way to try to apply the heat bath approach to generate a random independent set in a graph G (that is, a subset no two of whose vertices are joined by an edge) is first to color the vertices so that no two adjacent vertices are the same color, and then, cycling through the color classes, replace the current independent set I by the union of $I \setminus K$ with some random subset of K, where K is the set of vertices of a particular color that are not joined by an edge to any vertex in I. It is simple to show that for any fixed color, this update operation preserves the uniform distribution on the set of independent sets, since it is nothing more than a random step in an edge-subgraph whose components are all degree-regular graphs (hypercubes, in fact). Since the composite mapping obtained by cycling through all the vertices gives an ergodic Markov chain, there is at most one stationary distribution, and the uniform distribution must be it.

Unfortunately, we do not know of any rigorous estimates for the rate at which the preceding algorithm gives convergence to the uniform distribution, nor do we know of a way to use coupling-ideas to get empirical estimates of the mixing time. However, in the case where G is bipartite, a pretty trick of Kim, Shor, and Winkler [38] permits us to apply our methods. Specifically, let us suppose that the vertices of G have been classified as white and black, so that every edge joins vertices of opposite color. If we write the independent set I as $I_{\text{white}} \cup I_{\text{black}}$, then the set of independent sets becomes a distributive lattice if one defines the meet of I and I' as $(I_{\text{white}} \cap I'_{\text{white}}) \cup (I_{\text{black}} \cup I'_{\text{black}})$ and their join as $(I_{\text{white}} \cup I'_{\text{white}}) \cup (I_{\text{black}} \cap I'_{\text{black}})$. Hence one can sample from the uniform distribution on the set of independent sets in any finite bipartite graph.

This Markov chain may be slowly mixing for some graphs; for instance, in the case of the complete bipartite graph on $n + n$ vertices, the Markov chain is isomorphic to the slowly-mixing Markov chain mentioned in our earlier cautionary note. Kim, Shor, and Winkler consider a variant in which the different independent sets I need not have equal probability, but have probabilities proportional to $q^{|I|}$. For values of q below a certain critical threshold, they found that rapid coupling takes place; our methods would be applicable to the generation of random independent sets in the sub-critical regime.

4. Applications to Statistical Mechanics

When a physical system is in thermodynamic equilibrium, the probability that the system is in a given state σ is proportional to $e^{-E_\sigma/kT}$ where E_σ is the energy of state σ, T is the absolute temperature, and k is Boltzmann's constant. This probability distribution is known as the Gibbs distribution. In effect, kT is the standard unit of energy; when kT is large, the energies of the states are not significant, and all states are approximately equally likely, but when kT is small, the system is likely to be in a low-energy state. In some cases, a very small change in some parameter (such as the temperature) causes a significant change in the physical system. This phenomenon is called a *phase transition*. If changing the temperature caused the phase transition, then the temperature at the phase transition is called the *critical temperature*.

In the study of phase transitions, physicists often consider idealized models of substances. Phase-transition phenomena are thought to fall into certain *universality classes*, so that if a substance and a model belong to the same universality class, the global properties of the model correspond well to those of the real-world substance. For example, carbon dioxide, xenon, and brass are thought to belong to the same universality class as the three-dimensional Ising model (see [8] and [12] and references contained therein). Other models that have received much attention include the Potts model [63] (which generalizes the Ising model) and the related random cluster model [23].

In Monte Carlo studies of phase transitions, it is essential to generate many random samples of the state of a system. Sampling methods include the Metropolis algorithm and multi-grid versions of it. However, even after the Markov chain has run for a long time, it is not possible to tell by mere inspection whether the system has converged to a steady-state distribution or whether it has merely reached some metastable state. In many cases, we can use the method of coupled Markov chains to eliminate this problem and provide samples precisely according to the steady-state distribution.

In subsection 4.1 we review the definition of the Ising model and describe a simple algorithm for obtaining unbiased Ising samples that works well if the system is above (and not too close to) the critical temperature. We also define the Potts model, which generalizes the Ising model to the situation in which there are more than two possible spins.

In subsection 4.2 we consider the random cluster model. This model turns out to be very useful, in large part because random Ising and Potts states can be derived from random-cluster states. In many cases the best way to get Ising or Potts states is via the random cluster model. We show how to apply the method of monotone coupling-from-the-past to get unbiased samples, and include a picture of a perfectly equilibrated Ising state obtained by this method (Figure 1).

Finally, in subsection 4.3 we show how to apply our methods to the square ice model and the dimer model on the square and honeycomb grids.

We also mention that interest in these models is not restricted to physicists. For instance, in image processing, to undo the effects of blurring and noise one may place a Gibbs distribution on the set of possible images. The energy of a possible image depends on the observed pixel values, and nearby pixels tend to have similar values. Sampling from this Gibbs distribution can be effective in reducing noise. See [26] and [9] for more information.

4.1. The Ising model

Here we introduce the Ising model and the single-site heat bath algorithm for sampling from it. The efficiency of the heat bath algorithm has been the object of much study [56] [53] [25] [33] [49] [46]. To summarize, it runs quickly at temperatures above the critical temperature, but below this temperature it takes an enormously long time to randomize. (In subsection 4.2 we will describe a different algorithm which does not suffer from this "critical slowing down"; however, that algorithm does not apply when different parts of the substance are subjected to magnetic fields of different polarity, which makes that algorithm less suitable for some applications, such as image processing.)

The Ising model was introduced to model ferromagnetic substances; it is also equivalent to a lattice gas model [8]. An Ising system consists of a collection of n small interacting magnets, possibly in the presence of an external magnetic field. Each magnet may be aligned up or down. (In general there are more directions that a magnet may point, but in crystals such as $FeCl_2$ and $FeCO_3$ there are in fact just two directions [8].) Magnets that are close to each other prefer to be aligned in the same direction, and all magnets prefer to be aligned with the external magnetic field (which sometimes varies from site to site, but is often constant). These preferences are quantified in the total energy E of the system

$$E = -\sum_{i<j} \alpha_{i,j}\sigma_i\sigma_j - \sum_i B_i\sigma_i,$$

where B_i is the strength of the external field as measured at site i, σ_i is 1 if magnet i is aligned up and -1 if magnet i is aligned down, and $\alpha_{i,j} \geq 0$ represents the interaction strength between magnets i and j.

Often the n magnets are arranged in a 2D or 3D lattice, and $\alpha_{i,j}$ is 1 if magnets i and j are adjacent in the lattice, and 0 otherwise.

Characterizing what the system looks like at a given temperature is useful in the study of ferromagnetism. To study the system, we may sample a random state from the Gibbs distribution with a Markov chain. The single-site heat bath algorithm, also known as Glauber dynamics, iterates the following operation: Pick a magnet, either in sequence or at random, and then randomize its alignment, holding all of the remaining magnets fixed. There are two possible choices for the next state, denoted by $\sigma_\uparrow$ and $\sigma_\downarrow$, with energies $E_\uparrow$ and $E_\downarrow$. We have $\mathrm{Prob}[\sigma_\uparrow]/\mathrm{Prob}[\sigma_\downarrow] = e^{-(E_\uparrow - E_\downarrow)/kT} = e^{-(\Delta E)/kT}$. Thus a single update is simple to perform, and it is easy to check that this defines an ergodic Markov chain for the Gibbs distribution.

The picure of the 4200x4200 Ising state has been left out of this copy of the paper. The sheer size of the figure was causing problems for some printers, and was a problem for people without high-bandwidth internet connections.
The original version containing this Ising state is still available upon request, send us email and we'll send it.

Figure 1: An equilibrated Ising state at the critical temperature on a 4200×4200 toroidal grid.

To get an exact sample, we make the following observation: If we have two spin-configurations σ and τ with the property that each spin-up site in σ is also spin-up in τ, then we may evolve both configurations simultaneously according to the single-site heat bath algorithm, and this property is maintained. The all-spins-up state is "maximal", and the all-spins-down state is "minimal", so we have a monotone Markov chain to which we can apply the method of coupling from the past. Indeed, this is just a special case of our general algorithm for attractive spin-systems.

It is crucial that the $\alpha_{i,j}$'s be non-negative; if this were not the case, the system would not be attractive, and our method would not apply. (A few special cases, such as a paramagnetic system on a bipartite lattice, reduce to the attractive case.) However, there are no additional constraints on the $\alpha_{i,j}$'s and the B_i's; once the system is known to be attractive, we can be sure that our method applies, at least in a theoretical sense.

Also note that we can update two spins in parallel if the corresponding sites are non-adjacent (i.e., if the associated $\alpha_{i,j}$ vanishes), because the spin at one such site does not affect the conditional distribution for the spins at another. For instance, on a square grid, we can update half of the spins in parallel in a single step, and then update the other half at the next step. Despite the speed-up available from parallelization, there is no guarantee that the heat bath Markov chain will be a practical one. Indeed, it is well-known that it becomes disastrously slow near the critical temperature.

An important generalization of the Ising model is the q-state Potts model, in which each site may have one of q different "spins". Wu [63] gives a survey describing the physical significance of the Potts model. In a Potts configuration σ, each site i has spin σ_i which is one of $1, 2, 3, \ldots, q$. The energy of a Potts configuration is

$$E = \sum_{i<j} \alpha_{i,j}(1 - \delta_{\sigma_i,\sigma_j}) + \sum_i B_i(1 - \delta_{\sigma_i,e_i}),$$

where δ is the Kronecker delta-function, equal to 1 if its subscripts are equal and 0 otherwise, $\alpha_{i,j} \geq 0$ is the interaction strength between sites i and j, B_i is the strength of the magnetic field at site i, and e_i is the polarity of the magnetic field at site i. As before, adjacent sites prefer to have the same spin. When $q = 2$, the Potts-model energy reduces to the Ising-model energy aside from an additive constant (which does not affect the Gibbs distribution) and a scaling factor of two (which corresponds to a factor of two in the temperature).

4.2. Random cluster model

The random cluster model was introduced by Fortuin and Kasteleyn [23] and generalizes the Ising and Potts models. The random cluster model is also closely related to the Tutte polynomial of a graph (see [11]). The Ising state shown in Figure 1 was generated with the methods described here.

In the random cluster model we have an undirected graph G, and the states of the system are subsets H of the edges of the graph. Often G is a two- or three-dimensional lattice. Each edge $\{i, j\}$ has associated with it a number p_{ij} between 0 and 1 indicating how likely the edge is to be in the subgraph. There is a parameter q which indicates how favorable it is for the subgraph to have many connected components. In particular, the probability of observing a particular subgraph $H \subseteq G$ is proportional to

$$\left(\prod_{\{i,j\}\in H} p_{ij}\right)\left(\prod_{\{i,j\}\notin H} (1 - p_{ij})\right) q^{\mathcal{C}(H)},$$

where $\mathcal{C}(H)$ is the number of connected components of H (isolated vertices count as components of size 1). To derive a random q-spin Potts state from a random H, one assigns a common random spin to all the vertices lying in any given connected component (see Sokal's survey [51] for more information on this).

Sweeny [54] used the results of Fortuin and Kasteleyn to generate random q-spin Potts states near the critical temperature. He used the "single-bond heat bath" algorithm (i.e. Glauber dynamics) to sample from the random cluster model, and then converted these samples into Potts states. The single-bond heat bath algorithm for sampling from the random cluster model is a Markov chain which focuses on a single edge of G at a time and, conditioning on the rest of H, randomly determines whether or not to include this edge in the new state H'. It turns out that the random clusters are in some ways more directly informative than the Potts states themselves, since for instance they can be used to obtain more precise information on spin correlations.

Swendsen and Wang [55] proposed a different Markov chain based on the relation between the random cluster and Potts models. Given a random-cluster sample, they compute a random Potts state consistent with the clusters. Given a Potts state, they compute a random subgraph of G consistent with the Potts state. Their chain alternates between these two phases. In another variation due to Wolff [62], a single cluster is built from the spin states and then flipped. One major advantage of these approaches over the single-bond heat bath algorithm is that they obviate the need to determine connectivity for many pairs of vertices adjacent in V. Determining the connectivity can be computationally expensive, so Sweeny gave an algorithm for dynamically keeping track of which vertices remain connected each time an edge is added or deleted. The dynamic connectivity algorithm is limited to planar graphs, but the Markov chain itself is general.

We will now argue that the single-bond heat bath algorithm is monotone for $q \geq 1$, so that the method of monotone coupling-from-the-past applies to it. Here the states of the Markov chain are partially ordered by subgraph-inclusion, the top state is G, and the bottom state is the empty graph. We claim that when $q \geq 1$, this partial order is preserved by the Markov chain. At each step we pick an edge (either randomly or in sequence) and do the heat bath to this particular edge. If the two sites connected by this edge are in separate clusters in both states, or in the same cluster in both states, then the probability of including the edge is the same, so the partial order is preserved. If in one state they are in the same cluster while in the second state they are in separate clusters, then 1) the first state is the larger state (in the partial order), and 2) the probability of putting a bond there is larger for the firsstate. Hence the partial order is preserved, and our approach can be applied.

We have been unable to find monotonicity in the Swendsen-Wang or Wolff algorithms. Recent developments in dynamic connectivity algorithms [30] may make the single-bond heat bath algorithm a viable option, given that one can obtain exact samples at effectively all temperatures simultaneously using the heat bath algorithm. (See the discussion at the end of subsection 3.1.)

Our implementation of the monotone coupling-from-the-past version of the single-bond heat bath algorithm, in both single-temperature and omnithermal versions (see subsection 3.1), appears to work well in practice, provided that q is not too large. For instance, when $q = 2$ on a 512×512 toroidal grid, at the critical temperature it is only necessary to start at around time -30 to get coalescence by time 0. (Here one step, also called a sweep, consists of doing a single-bond heat bath step for each edge of the graph.) Using a naive connectivity algorithm, each sweep takes about twenty seconds on a Sparcstation. Omnithermal sweeps (see subsection 3.1) take longer, in part because determining connectivity is somewhat more involved, but principally because in the single-temperature case a large fraction of the heat bath steps do not require a connectivity query, whereas in the omnithermal case they all do. Since nearly all of the computer's time is spent computing connectivity, the incentive for improving on current dynamic connectivity algorithms is clear.

The preliminary omnithermal results are striking; in Figure 2 one can see quite clearly the critical point, and how the internal energy and spontaneous magnetization vary with temperature on the three-dimensional grid when $q = 2$. Other macroscopically-defined quantities may also be graphed as a monotone function of p. This makes it possible to obtain estimates of the critical exponents from just a single (omnithermal) sample, whereas other approaches require a large number of samples at a number of different values of p close to criticality. We are hopeful that exact omnithermal sampling will be of use with Monte Carlo studies of the Ising and Potts models.

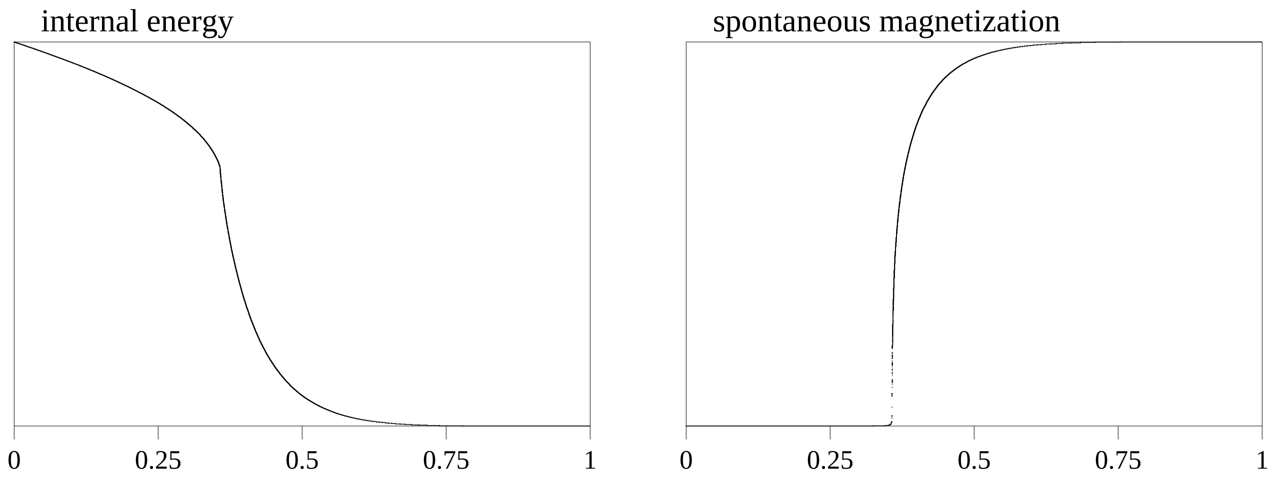

Figure 2: The internal energy and spontaneous magnetization of an Ising system as a function of $p = 1 - e^{-\Delta E/kT}$ on the $100 \times 100 \times 100$ grid with periodic boundary conditions. Data points are given for 2^{32} distinct values of the temperature, and are the result of a single simulation.

4.3. Ice and dimer models

In the square ice (or "six-vertex") model whose residual entropy was determined by Lieb [40], states are assignments of orientation to the edges of a square grid, such that at each internal vertex there are equal numbers of incoming and outgoing edges (the "ice condition"). The six vertex-configurations correspond to the six ways in which a water molecule in an ice crystal can orient itself so that its two protons (hydrogen atoms) are pointing towards two of the four adjacent oxygen atoms, and the ice condition reflects the fact that the protons from adjacent molecules will repel each other. The assumption that the ice condition is satisfied everywhere is tantamount to the assumption that the system is at temperature zero, so that in particular the Gibbs distribution is just the uniform distribution on the set of minimum-energy configurations.

Let us assume that our square grid is a finite rectangle, where sites along the boundary have edges that lead nowhere but nonetheless have a definite orientation. To turn ice-configurations on this rectangle into elements of a distributive lattice, we first pass to the dual model by rotating each directed edge by 90 degrees about its midpoint. This gives a model on the dual square grid in which every internal square cell must be bounded by two clockwise edges and two counterclockwise edges. One may think of such an orientation as a "discrete conservative vector field" on the set of edges of the grid; here conservativity means that if one travels between two points in the grid, the number of forward-directed edges along which one travels, minus the number of backward-directed edges, is independent of the path one takes between the two points.

One can then introduce an integer-valued function on the vertices, called a *height function*, with the property that adjacent vertices have heights that differ by 1, such that the edge between vertex i and vertex j points from i to j if and only if j is the higher of the two vertices. (If one pursues the analogue with field theory, one might think of this as a potential function associated with the discrete vector field.) Height functions for the six-vertex model were first introduced by van Beijeren [58].

The set of height functions with prescribed boundary conditions can be shown to form a distributive lattice under the natural operations of taking the maximum or minimum of the heights of two height functions at all the various points in the grid [47]. Hence one can sample from the uniform distribution on the set of such height functions, and thereby obtain a random sample from the set of ice-configurations.

Note that this approach applies to models other than square ice; for instance, it also applies to the "twenty-vertex" model on a triangular grid, in which each vertex has three incoming edges and three outgoing edges. However, in all these applications it is important that the finite graph that one uses be planar. In particular, one cannot apply these ideas directly to the study of finite models with free periodic boundary conditions (or equivalently graphs on a torus), because such graphs are not in general planar.

Another class of models to which our methods apply are dimer models on bipartite planar graphs. A dimer configuration on such a graph is a subset of the edges such that every vertex of the graph belongs to exactly one of the chosen edges. This model corresponds to adsorption of diatomic molecules on a crystal surface, and the assumption that every vertex belongs to an edge corresponds to the assumption that the system is in a lowest-energy state, as will be the case at zero temperature.

Here, as in the case of ice-models, one can introduce a height function that encodes the combinatorial structure and makes it possible to view the states as elements of a distributive lattice. This approach can be traced back to Levitov [39] and Zheng and Sachdev [64] in the case of the square lattice, and to Blöte and Hilhorst [14] in the case of the hexagonal lattice. An independent development is due to Thurston [57], building on earlier work of Conway [17]. A generalization of this technique is described in [47].

Thurston's article describes the construction not in terms of dimer models (or equivalently perfect matchings of graphs) but rather in terms of tilings. That is, a dimer configuration on a square or hexagonal grid corresponds to a tiling of a plane region by dominoes (unions of two adjacent squares in the dual grid) or lozenges (unions of two adjacent equilateral triangles in the dual grid).

Using the distributive lattice structure on the set of tilings, and applying coupling-from-the-past, one can generate random tilings of finite regions. This permits us to study the effects that the imposition of boundary conditions can have even well away from the boundary. One such phenomenon is the "arctic circle effect" described in [35]; using our random-generation procedure we have been able to ascertain that such domain-wall effects are fairly general, albeit associated with boundary conditions that may be deemed non-physical.

For instance, Figure 3 shows a particular finite region and a tiling chosen uniformly at random from the set of all domino-tilings of that region. To highlight the non-homogeneity of the statistics of the tiling, we have shaded those horizontal dominoes whose left square is black, under the natural black/white checkerboard coloring of the squares. The heterogeneity of the statistics is a consequence of the "non-flatness" of the height function on the boundary; this phenomenon is given a more detailed study in [16]. The tiling was generated using software written by the first author's undergraduate research assistants, using the methods described in this article.

Computer programs for generating random configurations of this kind can used as exploratory tools for the general problem of understanding the role played by boundary conditions in these classes of models.

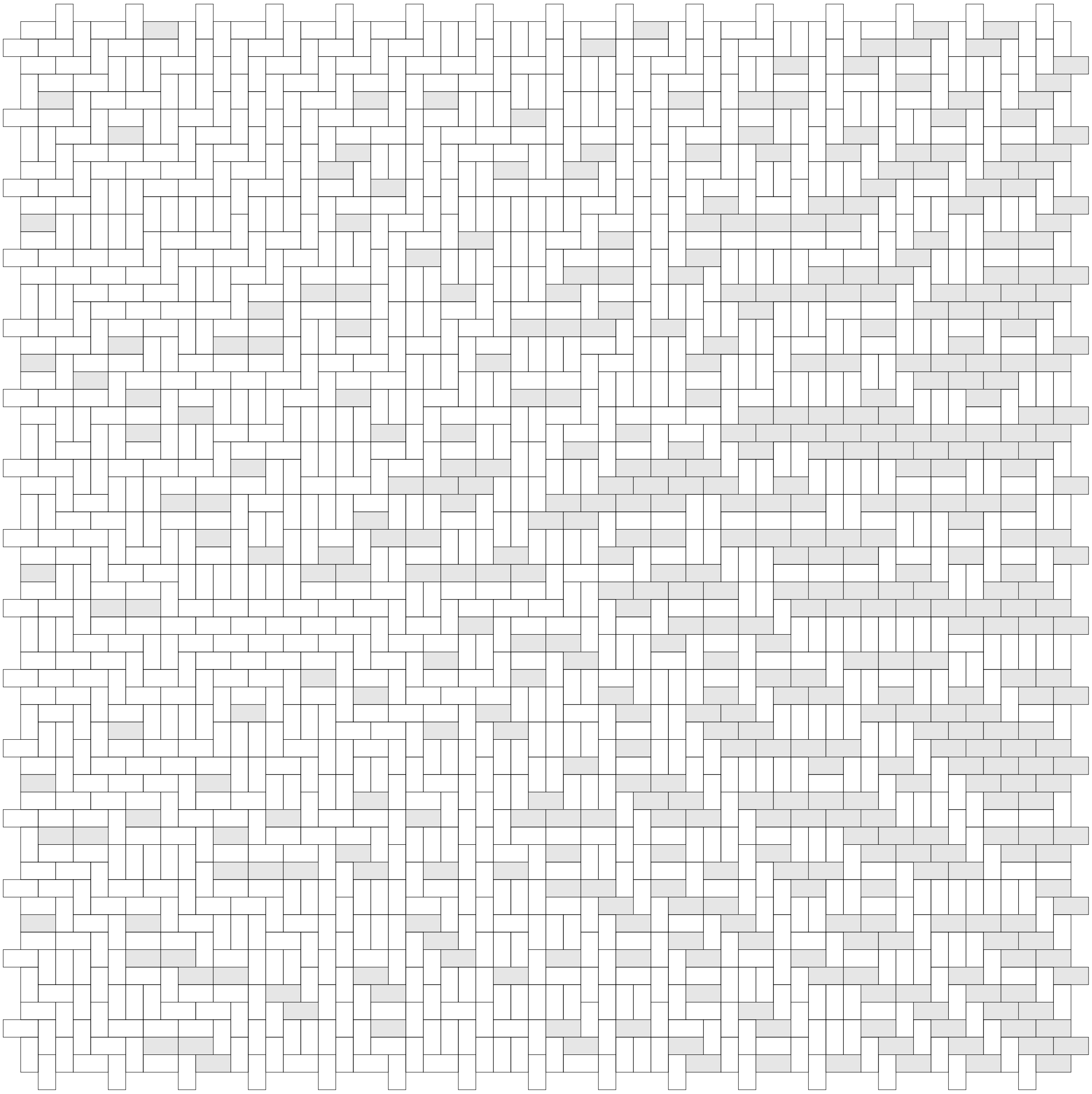

Figure 3: A random tiling of a finite region by dominoes.

5. Running Time

5.1. Time-to-coalescence

In this section we bound the coupling time of a monotone Markov chain, i.e. a chain with partially ordered state space whose moves preserve the partial order. These bounds directly relate to the running time of our exact sampling procedure. We bound the expected run time, and the probability that a run takes abnormally long, in terms of the mixing time of the Markov chain. If the underlying monotone Markov chain is rapidly mixing, then it is also rapidly coupling, so that there is essentially no reason not to apply coupling-from-the-past when it is possible to do so. If the mixing time is unknown, then in it may be estimated from the coupling times. We also bound the probability that a run takes much longer than these estimates.

Recall that the random variable T_* is the smallest t such that $F_{-t}^0(\hat{0}) = F_{-t}^0(\hat{1})$. Define T^*, the time to coalescence, to be the smallest t such that $F_0^t(\hat{0}) = F_0^t(\hat{1})$. Note that $\Pr[T_* > t]$, the probability that $F_{-t}^0(\cdot)$ is not constant, equals the probability that $F_0^t(\cdot)$ is not constant, $\Pr[T^* > t]$. The running time of the algorithm is linear in T_*, but since T_* and T^* are governed by the same probability distribution, in this section we focus on the conceptually simpler T^*.

In order to relate the coupling time to the mixing time, we will consider three measures of progress towards the steady state distribution π: $\mathrm{Exp}[T^*]$, $\mathrm{Prob}[T^* > K]$ for particular or random K, and $\overline{d}(k) = \max_{\mu_1,\mu_2} \|\mu_1^k - \mu_2^k\|$ for particular k, where μ^k is the distribution governing the Markov chain at time k when started at time 0 in a random state governed by the distribution μ. Let ρ_0 and ρ_1 be the distributions on the state space S that assign probability 1 to $\hat{0}$ and $\hat{1}$, respectively.

Theorem 5 *Let l be the length of the longest chain (totally ordered subset) in the partially ordered state-space S. Then*

$$\frac{\mathrm{Prob}[T^* > k]}{l} \leq \overline{d}(k) \leq \mathrm{Prob}[T^* > k].$$

Proof: If x is an element of the ordered state-space S, let $h(x)$ denote the length of the longest chain whose top element is x. Let the random variables X_0^k and X_1^k denote the states of (the two copies of) the Markov chain after k steps when started in states $\hat{0}$ and $\hat{1}$, respectively. If $X_0^k \neq X_1^k$ then $h(X_0^k) + 1 \leq h(X_1^k)$ (and if $X_0^k = X_1^k$ then $h(X_0^k) = h(X_1^k)$). This yields

$$\begin{aligned}
\mathrm{Prob}[T^* > k] &= \mathrm{Prob}[X_0^k \neq X_1^k] \\
&\leq E[h(X_1^k) - h(X_0^k)] \\
&= \left| E_{\rho_1^k}[h(X)] - E_{\rho_0^k}[h(X)] \right| \\
&\leq \|\rho_1^k - \rho_0^k\| \left[\max_x h(x) - \min_x h(x)\right] \\
&\leq \overline{d}(k)\, l,
\end{aligned}$$

proving the first inequality. To prove the second, consider a coupling in which one copy of the chain starts in some distribution μ_1 and the other starts in distribution μ_2. By the monotonicity of the coupling, the probability that the two copies coalesce within k steps is at least $\mathrm{Prob}[T^* \leq k]$. Hence our coupling achieves a joining of μ_1^k and μ_2^k such that the two states disagree only on an event of probability at most $\mathrm{Prob}[T^* > k]$. It follows that the total variation distance between the two distributions is at most $\mathrm{Prob}[T^* > k]$. □

Next we show that $\mathrm{Prob}[T^* > k]$ is submultiplicative. At this point we will assume that the valid moves at each time are the same. So for instance, if a random process operates on red vertices and then on blue vertices, these two operations together are considered one step.

Theorem 6 *Let K_1 and K_2 be nonnegative integer random variables (which might be constant). Then*

$$\mathrm{Prob}[T^* > K_1 + K_2] \leq \mathrm{Prob}[T^* > K_1] \cdot \mathrm{Prob}[T^* > K_2].$$

Proof: The event that $F_0^{K_1}$ is constant and the event that $F_{K_1}^{K_1+K_2}$ is constant are independent, and if either one is constant, then $F_0^{K_1+K_2}$ is constant. □

Next we estimate tail-probabilities for T^* in terms of the expected value of T^*, and vice versa.

Lemma 7

$$k\,\mathrm{Prob}[T^* > k] \leq \mathrm{Exp}[T^*] \leq k/\mathrm{Prob}[T^* \leq k].$$

Proof: The first inequality follows from the non-negativity of T^*. To prove the second, note that if we put $\varepsilon = \mathrm{Prob}[T^* > k]$, then by submultiplicativity, $\mathrm{Prob}[T^* > ik] \leq \varepsilon^i$. Hence $\mathrm{Exp}[T^*] \leq k + k\varepsilon + k\varepsilon^2 + \cdots = k/\mathrm{Prob}[T^* \leq k]$. □

Now we can say what we meant by "if the Markov chain is rapidly mixing then it is rapidly coupling" in the first paragraph of this section. The mixing time threshold T_{mix} is defined to be the smallest k for which $\overline{d}(k) \leq 1/e$ [5]. Let l be the length of the longest chain in the partially ordered state space. Since $\overline{d}(k)$ is submultiplicative (see [2]), after $k = T_{\mathrm{mix}}(1 + \ln l)$ steps, $\overline{d}(k) \leq 1/el$, so $\mathrm{Prob}[T^* > k] \leq 1/e$ by Theorem 5, and by Lemma 7,

$$\mathrm{Exp}[T^*] \leq k/(1 - 1/e) < 2k = 2T_{\mathrm{mix}}(1 + \ln l).$$

It has been noted [18] that many Markov chains exhibit a sharp threshold phenomenon: after $(1-\varepsilon)T_{\mathrm{mix}}$ steps the state is very far from being random, but after $(1+\varepsilon)T_{\mathrm{mix}}$ steps the state is very close to being random (i.e. $\overline{d}(k)$ is close to 0). In such chains the coupling time will be less than $O(T_{\mathrm{mix}} \log l)$.

In addition to being useful as a source of random samples, our method can also be used as a way of estimating the mixing time of a random walk. One application to which this might be put is the retrospective analysis of someone else's (or one's own) past simulations, which were undertaken with only the experimenter's intuitive sense to guide the choice of how long the chain needed to be run before the initialization bias would be acceptably small. Using our technique, one can now assess whether the experimenter's intuitions were correct.

For instance, suppose one takes ten independent samples of the coupling time random variable T^*, and obtains $10T_{\mathrm{est}} = T_1 + \cdots + T_{10} = 997$. Then one can be fairly confident that if one were to run the Markov chain from an arbitrary starting point for 1000 steps (or if someone had done so in the past), the residual initialization bias would be less than 2^{-10}. Assertions of this kind can be made rigorous. Specifically, we can argue that if one treats T_{est} as a random variable, then the initialization bias of the Markov chain when run for random time $10T_{\mathrm{est}}$ is at most 2^{-10}. By symmetry we have $\mathrm{Prob}[T^* > T_i] \leq 1/2$, and then by submultiplicativity we get

$$\mathrm{Prob}[T^* > T_1 + \cdots + T_{10}] \leq 2^{-10}.$$

Since $\overline{d}(k)$ is bounded by $\mathrm{Prob}[T^* > k]$, the initialization bias is at most 2^{-10}. (We cannot make any such rigorous assertion if we condition on the event $10T_{\mathrm{est}} = 997$, nor should we expect to be able to do so in the absence of more detailed information about the nature of the Markov chain.)

Another approach would be to consider the maximum coupling time of a number of runs.

Theorem 8 *Let $T_1, \ldots, T_m$ and T^* be independent samples of the coupling time. Then*

$$\mathrm{Prob}[T^* > j \max(T_1, \ldots, T_m)] \leq \frac{j!m!}{(j+m)!}.$$

Proof: Let $T_{\max}$ denote $\max(T_1, \ldots, T_m)$. We will construct coupling times $S_1, \ldots, S_j$ and T^* such that $T^*, T_1, \ldots, T_m$ are mutually independent and $T_1, \ldots, T_m, S_1, \ldots, S_j$ are mutually independent, but $S_1, \ldots, S_j$ and T^* are dependent. Each coupling time is determined by the random variables (earlier called U_t) that are used in simulating the Markov chain; call these random variables "moves". Let the first $T_{\max}$ of the moves for the time S_i be used for moves $(i-1)T_{\max}$ through $iT_{\max} - 1$ of time T^*, and let the remaining random moves for S_i be independent of the other random moves. If for any i we have $S_i \le T_{\max}$, then moves $(i-1)T_{\max}$ through $iT_{\max} - 1$ of time T^* are coalescing, whence $T^* \le jT_{\max}$. Thus, $\text{Prob}[T^* > jT_{\max}] \le \text{Prob}[S_i > T_{\max},\ i = 1, \ldots, j] = \text{Prob}[S_i > T_k,\ i = 1, \ldots, j,\ k = 1, \ldots, m]$. But if we have $j + m$ i.i.d. random variables, the probability that the last j are strictly larger than the first first m is at most $1/\binom{j+m}{j}$. □

In the above example with $m = 10$, if we take $j = 6$ then $\text{Prob}[T^* > 6T_{\max}] \le 1/8008$. If the longest of the ten runs takes 150 steps, then the randomized upper bound $6T_{\max}$ is 900. Calculations of this sort clearly can help an experimenter determine when her initialization bias is likely to be acceptably small.

5.2. Optimizing performance

As we mentioned in subsection 2.2, when applying coupling-from-the-past in the context of monotone Monte Carlo algorithms, it would be grossly inefficient to consider F_{-T}^0 for each positive integer T. We may restrict T to take on the values $T_0 < T_1 < \cdots$. Earlier we recommended taking the simulation start-times to be $-T_i = -2^i$. We shall see that this choice is close to optimal.

Let T_* denote the minimum value of T for which $F_{-T}^0(\hat{0}) = F_{-T}^0(\hat{1})$. One natural choice is to take $T_1 = rT_0$, $T_2 = rT_1$, etc., for some initial trial value T_0 and some ratio r. Then the number of simulation steps required to find the value of $F_{-T_*}^0(\hat{0}) = F_{-T_*}^0(\hat{1})$ is $2T_0 + 2rT_0 + 2r^2T_0 + \cdots + 2r^kT_0$, where k is the least k such that $r^kT_0 \ge T_*$. (The factor of 2 comes from the fact that we must simulate using both $\hat{0}$ and $\hat{1}$ as initial states.) The number of required steps is

$$\frac{r^{k+1}-1}{r-1}2T_0 < \frac{r^2}{r-1}r^{k-1}2T_0 \le \frac{r^2}{r-1}2T_*$$

where in the second inequality we assumed $T_0 \le T_*$. On the other hand, if one could magically guess T_*, then computing $F_{-T_*}^0(\hat{0})$ and $F_{-T_*}^0(\hat{1})$ (and in so doing verifying that the value of T_* is no greater than claimed) would take a full $2T_*$ simulation steps. The ratio between the two — worst-case and best-case — is at most $r^2/(r-1)$, which is minimized at $r = 2$, where it takes the value 4. Hence, if our goal is to minimize the worst-case number of steps, we should take $r = 2$.

One might be concerned that it is better to minimize the expected number of steps, rather than the worst case. Indeed, we argue that to minimize the expected number of steps one should pick $r = e$ $(= 2.71828\cdots)$. Let u be the fractional part of $\log_r(T_0/T_*)$. It is a reasonable heuristic to suppose that u is approximately uniformly distributed. In fact, by randomizing the choice of T_0, we can force this assumption to be true. In any case, the number of simulation steps needed to find the unbiased sample is

$$2r^uT_* + 2r^{u-1}T_* + 2r^{u-2}T_* + \cdots < 2\frac{r^u}{1-1/r}T_*.$$

The expected value of r^u is $(r-1)/\ln r$, so the expected number of steps is bounded above by $2T_*r/\ln r$. To minimize $r/\ln r$ we set $r = e$, with the expected number of steps equal to $2eT_* \approx 2.72(2T_*)$.

In practice, we do not expect to make great savings in time from this randomization strategy. Indeed, using $r = 2$ we find that the expected number of steps required is approximately 2.89 times

$2T_*$, which is quite close to the optimal achieved by $r = e$. Hence we have adopted the simpler doubling-procedure in our article.

Acknowledgements

We thank David Aldous, Persi Diaconis, and Jim Fill for their useful suggestions.

References

[1] David Aldous. On simulating a Markov chain stationary distribution when transition probabilities are unknown, 1994. Preprint.

[2] David Aldous and Persi Diaconis. Strong uniform times and finite random walks. *Advances in Applied Mathematics*, 8(1):69–97, 1987.

[3] David Aldous, László Lovász, and Peter Winkler. Fast mixing in a Markov chain, 1995. In preparation.

[4] David J. Aldous. A random walk construction of uniform spanning trees and uniform labelled trees. *SIAM Journal of Discrete Mathematics*, 3(4):450–465, 1990.

[5] David J. Aldous and James A. Fill. *Reversible Markov Chains and Random Walks on Graphs.* Book in preparation.

[6] Noga Alon and Joel H. Spencer. *The Probabilistic Method.* John Wiley & Sons, Inc., 1992. With appendix by Paul Erdős.

[7] Søren Asmussen, Peter W. Glynn, and Hermann Thorisson. Stationary detection in the initial transient problem. *ACM Transactions on Modeling and Computer Simulation*, 2(2):130–157, 1992.

[8] Rodney J. Baxter. *Exactly Solved Models in Statistical Mechanics.* Academic Press, 1982.

[9] Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society* B, 48(3):259–302, 1986.

[10] Julian Besag and Peter Clifford. Generalized Monte Carlo significance tests. *Biometrika*, 76(4):633–642, 1989.

[11] Norman Biggs. *Algebraic Graph Theory.* Cambridge University Press, second edition, 1993.

[12] J. J. Binney, N. J. Dowrick, A. J. Fisher, and M. E. J. Newman. *The Theory of Critical Phenomena: An Introduction to the Renormalization Group.* Oxford University Press, 1992.

[13] Anders Björner. Orderings of Coxeter groups. In *Combinatorics and Algebra*, pages 175–195. American Mathematical Society, 1984. Contemporary Mathematics, #34.

[14] H. W. J. Blöte and H. J. Hilhorst. Roughening transitions and the zero-temperature triangular Ising antiferromagnet. *Journal of Physics* A, 15(11):L631–L637, 1982.

[15] Andrei Broder. Generating random spanning trees. In *Foundations of Computer Science*, pages 442–447, 1989.

[16] Henry Cohn, Noam Elkies, and James Propp. Local statistics for random domino tilings of the Aztec diamond. *Duke Mathematical Journal*, 1996. To appear.

[17] John Conway and Jeffrey Lagarias. Tiling with polyominoes and combinatorial group theory. *Journal of Combinatorial Theory, series A*, 53:183–208, 1990.

[18] Persi Diaconis. *Group Representations in Probability and Statistics*. Institute of Mathematical Statistics, 1988.

[19] Persi Diaconis and Laurent Saloff-Coste. What do we know about the Metropolis algorithm? In *ACM Symposium on the Theory of Computing*, pages 112–129, 1995.

[20] Persi Diaconis and Daniel Stroock. Geometric bounds for eigenvalues of Markov chains. *The Annals of Applied Probability*, 1(1):36–61, 1991.

[21] Persi Diaconis and Bernd Sturmfels. Algebraic algorithms for sampling from conditional distributions. *The Annals of Statistics*. To appear.

[22] Stefan Felsner and Lorenz Wernisch. Markov chains for linear extensions, the two-dimensional case, 1996. Manuscript.

[23] C. M. Fortuin and P. W. Kasteleyn. On the random cluster model. I. Introduction and relation to other models. *Physica*, 57(4):536–564, 1972.

[24] C. M. Fortuin, P. W. Kasteleyn, and J. Ginibre. Correlation inequalites on some partially ordered sets. *Communications in Mathematical Physics*, 22:89–103, 1971.

[25] Arnoldo Frigessi, Chii-Ruey Hwang, Shuenn-Jyi Sheu, and Patrizia di Stefano. Convergence rates of the Gibbs sampler, the Metropolis algorithm and other single-site updating dynamics. *Journal of the Royal Statistical Society* B, 55(1):205–219, 1993.

[26] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–724, 1984.

[27] David Griffeath. *Additive and Cancellative Interacting Particle Systems*. Springer-Verlag, 1979. Lecture Notes in Mathematics, #724.

[28] Geoffrey Grimmett. *Percolation*. Springer-Verlag, 1989.

[29] Geoffrey Grimmett. The stochastic random-cluster process and the uniqueness of random-cluster measures. *The Annals of Probability*, 23(4):1461–1510, 1995. Special invited paper.

[30] Monika Rauch Henzinger and Valerie King. Randomized dynamic algorithms with polylogarithmic time per operation. In *ACM Symposium on the Theory of Computing*, pages 519–527, 1995.

[31] Richard Holley. Remarks on the FKG inequalities. *Communications in Mathematical Physics*, 36:227–231, 1974.

[32] Richard Holley and Thomas Liggett. Ergodic theorems for weakly interacting systems and the voter model. *Annals of Probability*, 3:643–663, 1975.

[33] Salvatore Ingrassia. On the rate of convergence of the Metropolis algorithm and Gibbs sampler by geometric bounds. *The Annals of Applied Probability*, 4(2):347–389, 1994.

[34] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM Journal on Computing*, 18(6):1149–1178, 1989.

[35] William Jockusch, James Propp, and Peter Shor. Random domino tilings and the arctic circle theorem, 1995. Preprint.

[36] Valen E. Johnson. Testing for convergence of Markov chain Monte Carlo algorithms using parallel sample paths, 1995. Preprint.

[37] Maurice G. Kendall. *Rank Correlation Methods*. Hafner Publishing Company, third edition, 1962.

[38] Jeong Han Kim, Peter Shor, and Peter Winkler. Personal communication.

[39] L. S. Levitov. Equivalence of the dimer resonating-valence-bond problem to the quantum roughening problem. *Physical Review Letters*, 64(1):92–94, 1990.

[40] Elliott Lieb. Residual entropy of square ice. *Physical Review*, 162:162–172, 1967.

[41] Thomas Liggett. *Interacting Particle Systems*. Springer-Verlag, 1985.

[42] Torgny Lindvall. *Lectures on the Coupling Method*. John Wiley & Sons, Inc., 1992.

[43] László Lovász and Miklós Simonovits. On the randomized complexity of volume and diameter. In *Foundations of Computer Science*, pages 482–491, 1992.

[44] László Lovász and Peter Winkler. Exact mixing in an unknown Markov chain. *Electronic Journal of Combinatorics*, 2, 1995. Paper #R15.

[45] Michael Luby, Dana Randall, and Alistair Sinclair. Markov chain algorithms for planar lattice structures (extended abstract). In *Foundations of Computer Science*, pages 150–159, 1995.

[46] F. Martinelli, E. Olivieri, and R. H. Schonmann. For 2-d lattice spin systems weak mixing implies strong mixing. *Communications in Mathematical Physics*, 165(1):33–47, 1994.

[47] James Propp. Lattice structure for orientations of graphs, 1993. Preprint.

[48] Jeffrey S. Rosenthal. Minorization conditions and convergence rates for Markov chain Monte Carlo. *Journal of the American Statistical Association*, 90(430):558–566, 1995.

[49] Roberto H. Schonmann. Slow droplet-driven relaxation of stochastic Ising models in the vicinity of the phase coexistence region. *Communications in Mathematical Physics*, 161(1):1–49, 1994.

[50] Alistair Sinclair. *Algorithms for Random Generation and Counting: A Markov Chain Approach*. Birkhäuser, 1993.

[51] Alan D. Sokal. Monte Carlo methods in statistical mechanics: Foundations and new algorithms, 1989. Lecture notes from Cours de Troisième Cycle de la Physique en Suisse Romande.

[52] Richard P. Stanley. *Enumerative Combinatorics*, volume 1. Wadsworth, Inc., 1986.

[53] Daniel W. Stroock and Boguslaw Zegarlinski. The logarithmic Sobolev inequality for discrete spin systems on a lattice. *Communications in Mathematical Physics*, 149(1):175–193, 1992.

[54] Mark Sweeny. Monte Carlo study of weighted percolation clusters relevant to the Potts models. *Physical Review* B, 27(7):4445–4455, 1983.

[55] Robert H. Swendsen and Jian-Sheng Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters*, 58(2):86–88, 1987.

[56] Lawrence E. Thomas. Bound on the mass gap for finite volume stochastic Ising models at low temperature. *Communications in Mathematical Physics*, 126(1):1–11, 1989.

[57] William Thurston. Conway's tiling groups. *American Mathematical Monthly*, 97:757–773, 1990.

[58] Henk van Beijeren. Exactly solvable model for the roughening transition of a crystal surface. *Physical Review Letters*, 38(18):993–996, 1977.

[59] David B. Wilson. Manuscript.

[60] David B. Wilson. Generating random spanning trees more quickly than the cover time. In *ACM Symposium on the Theory of Computing*, pages 296–303, 1996.

[61] David B. Wilson and James G. Propp. How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 448–457, 1996.

[62] Ulli Wolff. Collective Monte Carlo updating for spin systems. *Physical Review Letters*, 62(4):361–364, 1989.

[63] F. Y. Wu. The Potts model. *Reviews of Modern Physics*, 54(1):235–268, 1982.

[64] Wei Zheng and Subir Sachdev. Sine-Gordon theory of the non-Néel phase of two-dimensional quantum antiferromagnets. *Physical Review* B, 40:2704–2707, 1989.