Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms

Alan D. Sokal

Department of Physics
New York University
4 Washington Place
New York, NY 10003
USA

E-mail: `SOKAL@NYU.EDU`

Lectures at the Cargèse Summer School on
"Functional Integration: Basics and Applications"
September 1996

These notes are an updated version of lectures given at the Cours de Troisième Cycle de la Physique en Suisse Romande (Lausanne, Switzerland) in June 1989. We thank the Troisième Cycle de la Physique en Suisse Romande and Professor Michel Droz for kindly giving permission to reprint these notes.

Note to the Reader

The following notes are based on my course "Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms" given at the Cours de Troisième Cycle de la Physique en Suisse Romande (Lausanne, Switzerland) in June 1989, and on my course "Multi-Grid Monte Carlo for Lattice Field Theories" given at the Winter College on Multilevel Techniques in Computational Physics (Trieste, Italy) in January–February 1991.

The reader is warned that some of this material is out-of-date (this is particularly true as regards reports of numerical work). For lack of time, I have made no attempt to update the text, but I have added footnotes marked "**Note Added 1996**" that correct a few errors and give additional bibliography.

My first two lectures at Cargèse 1996 were based on the material included here. My third lecture described the new finite-size-scaling extrapolation method of [97, 98, 99, 100, 101, 102, 103].

1 Introduction

The goal of these lectures is to give an introduction to current research on Monte Carlo methods in statistical mechanics and quantum field theory, with an emphasis on:

1) the conceptual foundations of the method, including the possible dangers and misuses, and the correct use of statistical error analysis; and

2) new Monte Carlo algorithms for problems in critical phenomena and quantum field theory, aimed at reducing or eliminating the "critical slowing-down" found in conventional algorithms.

These lectures are aimed at a mixed audience of theoretical, computational and mathematical physicists — some of whom are currently doing or want to do Monte Carlo studies themselves, others of whom want to be able to evaluate the reliability of published Monte Carlo work.

Before embarking on 9 hours of lectures on Monte Carlo methods, let me offer a warning:

> *Monte Carlo is an extremely* bad *method; it should be used only when all alternative methods are worse.*

Why is this so? Firstly, *all* numerical methods are potentially dangerous, compared to analytic methods; there are more ways to make mistakes. Secondly, as numerical methods go, Monte Carlo is one of the least efficient; it should be used only on those intractable problems for which all other numerical methods are even *less* efficient.

Let me be more precise about this latter point. Virtually all Monte Carlo methods have the property that the statistical error behaves as

$$\text{error} \sim \frac{1}{\sqrt{\text{computational budget}}}$$

(or worse); this is an essentially universal consequence of the central limit theorem. It may be possible to improve the *proportionality constant* in this relation by a factor of 10^6 or more — this is one of the principal subjects of these lectures — but the overall $1/\sqrt{n}$ behavior is inescapable. This should be contrasted with traditional deterministic numerical methods whose rate of convergence is typically something like $1/n^4$ or e^{-n} or e^{-2^n}. Therefore, Monte Carlo methods should be used only on those extremely difficult problems in which all alternative numerical methods behave even *worse* than $1/\sqrt{n}$.

Consider, for example, the problem of numerical integration in d dimensions, and let us compare Monte Carlo integration with a traditional deterministic method such as Simpson's rule. As is well known, the error in Simpson's rule with n nodal points behaves asymptotically as $n^{-4/d}$ (for smooth integrands). In low dimension ($d < 8$) this is much better than Monte Carlo integration, but in high dimension ($d > 8$) it is much worse. So it is not surprising that Monte Carlo is the method of choice for performing high-dimensional integrals. It is still a bad method: with an error proportional to $n^{-1/2}$, it is difficult to achieve more than 4 or 5 digits accuracy. But numerical integration in high dimension is very difficult; though Monte Carlo is bad, all other known methods are worse.[1]

In summary, Monte Carlo methods should be used only when neither analytic methods nor deterministic numerical methods are workable (or efficient). One general domain of application of Monte Carlo methods will be, therefore, to systems with *many degrees of freedom, far from the perturbative regime*. But such systems are precisely the ones of greatest interest in statistical mechanics and quantum field theory!

It is appropriate to close this introduction with a general methodological observation, ably articulated by Wood and Erpenbeck [3]:

> ...these [Monte Carlo] investigations share some of the features of ordinary experimental work, in that they are susceptible to both statistical and systematic errors. With regard to these matters, we believe that papers should meet much the same standards as are normally required for experimental investigations. We have in mind the inclusion of estimates of statistical error, descriptions of experimental conditions (i.e. parameters of the calculation),

[1]This discussion of numerical integration is grossly oversimplified. Firstly, there are deterministic methods better than Simpson's rule; and there are also sophisticated Monte Carlo methods whose asymptotic behavior (on smooth integrands) behaves as n^{-p} with p strictly greater than 1/2 [1, 2]. Secondly, for all these algorithms (except standard Monte Carlo), the asymptotic behavior as $n \to \infty$ may be irrelevant in practice, because it is achieved only at ridiculously large values of n. For example, to carry out Simpson's rule with even 10 nodes per axis (a very coarse mesh) requires $n = 10^d$, which is unachievable for $d \gtrsim 10$.

relevant details of apparatus (program) design, comparisons with previous investigations, discussion of systematic errors, etc. Only if these are provided will the results be trustworthy guides to improved theoretical understanding.

2 Dynamic Monte Carlo Methods: General Theory

All Monte Carlo work has the same general structure: given some probability measure π on some configuration space S, we wish to generate many random samples from π. How is this to be done?

Monte Carlo methods can be classified as *static* or *dynamic*. Static methods are those that generate a sequence of *statistically independent* samples from the desired probability distribution π. These techniques are widely used in Monte Carlo numerical integration in spaces of not-too-high dimension [2]. But they are unfeasible for most applications in statistical physics and quantum field theory, where π is the Gibbs measure of some rather complicated system (extremely many coupled degrees of freedom).

The idea of *dynamic* Monte Carlo methods is to invent a *stochastic process* with state space S having π as its unique equilibrium distribution. We then simulate this stochastic process on the computer, starting from an arbitrary initial configuration; once the system has reached equilibrium, we measure time averages, which converge (as the run time tends to infinity) to π-averages. In physical terms, we are inventing a *stochastic time evolution* for the given system. Let us emphasize, however, that this time evolution *need not correspond to any real "physical" dynamics*: rather, the dynamics is simply a numerical algorithm, and it is to be chosen, like all numerical algorithms, on the basis of its computational efficiency.

In practice, the stochastic process is always taken to be a *Markov process*. So our first order of business is to review some of the general theory of Markov chains.[2] For simplicity let us assume that the state space S is discrete (i.e. finite or countably infinite). Much of the theory for general state space can be guessed from the discrete theory by making the obvious replacements (sums by integrals, matrices by kernels), although the proofs tend to be considerably harder.

Loosely speaking, a Markov chain (= discrete-time Markov process) with state space S is a sequence of S-valued random variables $X_0, X_1, X_2, \ldots$ such that successive transitions $X_t \to X_{t+1}$ are statistically independent ("the future depends on the past only through the present"). More precisely, a Markov chain is specified by two ingredients:

[2]The books of Kemeny and Snell [4] and Iosifescu [5] are excellent references on the theory of Markov chains with *finite* state space. At a somewhat higher mathematical level, the books of Chung [6] and Nummelin [7] deal with the cases of *countable* and *general* state space, respectively.

- The *initial distribution* α. Here α is a probability distribution on S, and the process will be defined so that $\mathbb{P}(X_0 = x) = \alpha_x$.
- The *transition probability matrix* $P = \{p_{xy}\}_{x,y\in S} = \{p(x \to y)\}_{x,y\in S}$. Here P is a matrix satisfying $p_{xy} \geq 0$ for all x, y and $\sum_y p_{xy} = 1$ for all x. The process will be defined so that $\mathbb{P}(X_{t+1} = y \,|\, X_t = x) = p_{xy}$.

The Markov chain is then completely specified by the joint probabilities

$$\mathbb{P}(X_0 = x_0,\, X_1 = x_1,\, X_2 = x_2,\, \ldots,\, X_n = x_n) \;\equiv\; \alpha_{x_0}\, p_{x_0x_1}\, p_{x_1x_2} \,\cdots\, p_{x_{n-1}x_n} \,. \tag{2.1}$$

This product structure expresses the fact that "successive transitions are independent".

Next we define the *n-step transition probabilities*

$$p_{xy}^{(n)} \;=\; \mathbb{P}(X_{t+n} = y \,|\, X_t = x) \,. \tag{2.2}$$

Clearly $p_{xy}^{(0)} = \delta_{xy}$, $p_{xy}^{(1)} = p_{xy}$, and in general $\{p_{xy}^{(n)}\}$ are the matrix elements of P^n.

A Markov chain is said to be *irreducible* if from each state it is possible to get to each other state: that is, for each pair $x, y \in S$, there exists an $n \geq 0$ for which $p_{xy}^{(n)} > 0$. We shall be considered almost exclusively with irreducible Markov chains.

For each state x, we define the *period of* x (denoted d_x) to be the greatest common divisor of the numbers $n > 0$ for which $p_{xx}^{(n)} > 0$. If $d_x = 1$, the state x is called *aperiodic*. It can be shown that, in an irreducible chain, all states have the same period; so we can speak of the chain having period d. Moreover, the state space can then be partitioned into subsets $S_1, S_2, \ldots, S_d$ around which the chain moves cyclically, i.e. $p_{xy}^{(n)} = 0$ whenever $x \in S_i$, $y \in S_j$ with $j - i \neq n \pmod d$. Finally, it can be shown that a chain is *irreducible and aperiodic* if and only if, for each pair x, y, there exists N_{xy} such that $p_{xy}^{(n)} > 0$ for *all* $n \geq N_{xy}$.

We now come to the fundamental topic in the theory of Markov chains, which is the problem of convergence to equilibrium. A probability measure $\pi = \{\pi_x\}_{x\in S}$ is called a *stationary distribution* (or *invariant distribution* or *equilibrium distribution*) for the Markov chain P in case

$$\sum_x \pi_x\, p_{xy} \;=\; \pi_y \qquad \text{for all } y. \tag{2.3}$$

A stationary probability distribution need not exist; but if it does, then a lot more follows:

Theorem 1 *Let P be the transition probability matrix of an* irreducible *Markov chain of period d. If a stationary probability distribution π exists, then it is unique, and $\pi_x > 0$ for all x. Moreover,*

$$\lim_{n\to\infty} p_{xy}^{(nd+r)} \;=\; \begin{cases} d\pi_y & \text{if } x \in S_i,\ y \in S_j \text{ with } j - i = r \pmod d \\ 0 & \text{if } x \in S_i,\ y \in S_j \text{ with } j - i \neq r \pmod d \end{cases} \tag{2.4}$$

for all x, y. In particular, if P is aperiodic, then

$$\lim_{n\to\infty} p_{xy}^{(n)} \;=\; \pi_y \,. \tag{2.5}$$

This theorem shows that the Markov chain converges as $t \to \infty$ to the equilibrium distribution π, irrespective of the initial distribution α. Moreover, under the conditions of this theorem much more can be proven — for example, a strong law of large numbers, a central limit theorem, and a law of the iterated logarithm. For statements and proofs of all these theorems, we refer the reader to Chung [6].

We can now see how to set up a dynamic Monte Carlo method for generating samples from the probability distribution π. It suffices to invent a transition probability matrix $P = \{p_{xy}\} = \{p(x \to y)\}$ satisfying the following two conditions:

(A) *Irreducibility.*[3] For each pair $x, y \in S$, there exists an $n \geq 0$ for which $p_{xy}^{(n)} > 0$.

(B) *Stationarity of* π. For each $y \in S$,

$$\sum_x \pi_x \, p_{xy} = \pi_y. \tag{2.6}$$

Then Theorem 1 (together with its more precise counterparts) shows that simulation of the Markov chain P constitutes a legitimate Monte Carlo method for estimating averages with respect to π. We can start the system in any state x, and the system is guaranteed to converge to equilibrium as $t \to \infty$ [at least in the averaged sense (2.4)]. Long-time averages of any observable f will converge with probability 1 to π-averages (strong law of large numbers), and will do so with fluctuations of size $\sim n^{-1/2}$ (central limit theorem). In practice we will discard the data from the initial transient, i.e. before the system has come close to equilibrium, but in principle this is not necessary (the bias is of order n^{-1}, hence asymptotically much smaller than the statistical fluctuations).

So far, so good! But while this is a *correct* Monte Carlo algorithm, it may or may not be an *efficient* one. The key difficulty is that the successive states X_0, X_1, ... of the Markov chain are *correlated* — perhaps very strongly — so the variance of estimates produced from the dynamic Monte Carlo simulation may be much higher than in static Monte Carlo (independent sampling). To make this precise, let $f = \{f(x)\}_{x \in S}$ be a real-valued function defined on the state space S (i.e. a real-valued observable) that is square-integrable with respect to π. Now consider the *stationary* Markov chain (i.e. start the system in the stationary distribution π, or equivalently, "equilibrate" it for a very long time prior to observing the system). Then $\{f_t\} \equiv \{f(X_t)\}$ is a stationary stochastic process with mean

$$\mu_f \equiv \langle f_t \rangle = \sum_x \pi_x \, f(x) \tag{2.7}$$

[3] We avoid the term "ergodicity" because of its multiple and conflicting meanings. In the physics literature, and in the mathematics literature on Markov chains with *finite* state space, "ergodic" is typically used as a synonym for "irreducible" [4, Section 2.4] [5, Chapter 4]. However, in the mathematics literature on Markov chains with *general* state space, "ergodic" is used as a synonym for "irreducible, aperiodic and positive Harris recurrent" [7, p. 114] [8, p. 169].

and *unnormalized autocorrelation function*[4]

$$\begin{aligned} C_{ff}(t) &\equiv \langle f_s f_{s+t} \rangle - \mu_f^2 \\ &= \sum_{x,y} f(x) \left[\pi_x p_{xy}^{(|t|)} - \pi_x \pi_y \right] f(y). \end{aligned} \tag{2.8}$$

The *normalized autocorrelation function* is then

$$\rho_{ff}(t) \equiv C_{ff}(t)/C_{ff}(0). \tag{2.9}$$

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

$$\tau_{exp,f} = \limsup_{t \to \infty} \frac{t}{-\log |\rho_{ff}(t)|} \tag{2.10}$$

and

$$\tau_{exp} = \sup_f \; \tau_{exp,f}. \tag{2.11}$$

Thus, τ_{exp} is the relaxation time of the slowest mode in the system. (If the state space is infinite, τ_{exp} might be $+\infty$!)

An equivalent definition, which is useful for rigorous analysis, involves considering the spectrum of the transition probability matrix P considered as an operator on the Hilbert space $l^2(\pi)$.[5] It is not hard to prove the following facts about P:

(a) The operator P is a contraction. (In particular, its spectrum lies in the closed unit disk.)

(b) 1 is a simple eigenvalue of P, as well as of its adjoint P^*, with eigenvector equal to the constant function $\mathbf{1}$.

(c) If the Markov chain is aperiodic, then 1 is the only eigenvalue of P (and of P^*) on the unit circle.

(d) Let R be the spectral radius of P acting on the orthogonal complement of the constant functions:

$$R \equiv \inf \left\{ r\colon\; \mathrm{spec}\,(P \restriction \mathbf{1}^\perp) \subset \{\lambda\colon |\lambda| \le r\} \right\}. \tag{2.12}$$

Then $R = e^{-1/\tau_{exp}}$.

Facts (a)–(c) are a generalized Perron-Frobenius theorem [9]; fact (d) is a consequence of a generalized spectral radius formula [10, Propositions 2.3–2.5].

[4] In the statistics literature, this is called the *autocovariance function*.

[5] $l^2(\pi)$ is the space of complex-valued functions on S that are square-integrable with respect to π: $\|f\| \equiv (\sum_x \pi_x |f(x)|^2)^{1/2} < \infty$. The inner product is given by $(f,g) \equiv \sum_x \pi_x f(x)^* g(x)$.

The rate of convergence to equilibrium from an initial nonequilibrium distribution can be bounded above in terms of R (and hence τ_{exp}). More precisely, let ν is a probability measure on S, and let us define its deviation from equilibrium in the l^2 sense,

$$d_2(\nu;\pi) \;\equiv\; ||\frac{\nu}{\pi} - 1||_{l^2(\pi)} \;=\; \sup_{||f||_{l^2(\pi)} \le 1} \left| \int f\, d\nu - \int f\, d\pi \right| \;. \tag{2.13}$$

Then, clearly,

$$d_2(\alpha P^t;\pi) \;\le\; ||P^t \restriction \mathbf{1}^\perp||\; d_2(\alpha;\pi)\;. \tag{2.14}$$

And by the spectral radius formula,

$$||P^t \restriction \mathbf{1}^\perp|| \;\sim\; R^t \;=\; \exp(-t/\tau_{exp}) \tag{2.15}$$

asymptotically as $t \to \infty$, with equality for all t if P is self-adjoint (see below).

On the other hand, for a given observable f we define the *integrated autocorrelation time*

$$\begin{aligned} \tau_{int,f} \;&=\; \frac{1}{2}\sum_{t=-\infty}^{\infty} \rho_{ff}(t) \\ &=\; \frac{1}{2} + \sum_{t=1}^{\infty} \rho_{ff}(t) \end{aligned} \tag{2.16}$$

[The factor of $\frac{1}{2}$ is purely a matter of convention; it is inserted so that $\tau_{int,f} \approx \tau_{exp,f}$ if $\rho_{ff}(t) \sim e^{-|t|/\tau}$ with $\tau \gg 1$.] The integrated autocorrelation time controls the statistical error in Monte Carlo measurements of $\langle f \rangle$. More precisely, the sample mean

$$\bar{f} \;\equiv\; \frac{1}{n}\sum_{t=1}^{n} f_t \tag{2.17}$$

has variance

$$\mathrm{var}(\bar{f}) \;=\; \frac{1}{n^2}\sum_{r,s=1}^{n} C_{ff}(r-s) \tag{2.18}$$

$$\;=\; \frac{1}{n}\sum_{t=-(n-1)}^{n-1} \left(1 - \frac{|t|}{n}\right) C_{ff}(t) \tag{2.19}$$

$$\;\approx\; \frac{1}{n}(2\tau_{int,f})\, C_{ff}(0) \quad \text{for } n \gg \tau \tag{2.20}$$

Thus, the variance of $\bar{f}$ is a factor $2\tau_{int,f}$ larger than it would be if the $\{f_t\}$ were statistically independent. Stated differently, the number of "effectively independent samples" in a run of length n is roughly $n/2\tau_{int,f}$.

It is sometimes convenient to measure the integrated autocorrelation time in terms of the equivalent pure exponential decay that would produce the same value of $\sum_{t=-\infty}^{\infty} \rho_{ff}(t)$, namely

$$\widetilde{\tau}_{int,f} \;\equiv\; \frac{-1}{\log\left(\frac{2\tau_{int,f}-1}{2\tau_{int,f}+1}\right)} \;. \tag{2.21}$$

This quantity has the nice feature that a sequence of uncorrelated data has $\tilde{\tau}_{int,f} = 0$ (but $\tau_{int,f} = \frac{1}{2}$). Of course, $\tilde{\tau}_{int,f}$ is ill-defined if $\tau_{int,f} < \frac{1}{2}$, as can occasionally happen in cases of anticorrelation.

In summary, the autocorrelation times τ_{exp} and $\tau_{int,f}$ play different roles in Monte Carlo simulations. τ_{exp} places an upper bound on the number of iterations n_{disc} which should be discarded at the beginning of the run, before the system has attained equilibrium; for example, $n_{disc} \approx 20\tau_{exp}$ is usually more than adequate. On the other hand, $\tau_{int,f}$ determines the statistical errors in Monte Carlo measurements of $\langle f \rangle$, once equilibrium has been attained.

Most commonly it is assumed that τ_{exp} and $\tau_{int,f}$ are of the same order of magnitude, at least for "reasonable" observables f. But this is *not* true in general. In fact, in statistical-mechanical problems near a critical point, one usually expects the autocorrelation function $\rho_{ff}(t)$ to obey a dynamic scaling law [11] of the form

$$\rho_{ff}(t;\beta) \;\sim\; |t|^{-a}\, F\left((\beta-\beta_c)\,|t|^b\right) \tag{2.22}$$

valid in the region

$$|t| \gg 1, \quad |\beta-\beta_c| \ll 1, \quad |\beta-\beta_c|\,|t|^b \text{ bounded.} \tag{2.23}$$

Here $a, b > 0$ are dynamic critical exponents and F is a suitable scaling function; β is some "temperature-like" parameter, and β_c is the critical point. Now suppose that F is continuous and strictly positive, with $F(x)$ decaying rapidly (e.g. exponentially) as $|x| \to \infty$. Then it is not hard to see that

$$\tau_{exp,f} \;\sim\; |\beta-\beta_c|^{-1/b} \tag{2.24}$$

$$\tau_{int,f} \;\sim\; |\beta-\beta_c|^{-(1-a)/b} \tag{2.25}$$

$$\rho_{ff}(t;\, \beta=\beta_c) \;\sim\; |t|^{-a} \tag{2.26}$$

so that $\tau_{exp,f}$ and $\tau_{int,f}$ have *different* critical exponents unless $a = 0$.[6] Actually, this should not be surprising: replacing "time" by "space", we see that $\tau_{exp,f}$ is the analogue of a correlation length, while $\tau_{int,f}$ is the analogue of a susceptibility; and (2.24)–(2.26) are the analogue of the well-known scaling law $\gamma = (2-\eta)\nu$ — clearly $\gamma \neq \nu$ in general! So it is crucial to distinguish between the two types of autocorrelation time.

Returning to the general theory, we note that one convenient way of satisfying condition (B) is to satisfy the following *stronger* condition:

$$\text{(B}'\text{)} \;\; \text{For each pair } x,y \in S,\;\; \pi_x p_{xy} = \pi_y p_{yx}. \tag{2.27}$$

[6] Our discussion of this topic in [12] is incorrect.

[Summing (B′) over x, we recover (B).] (B′) is called the *detailed-balance condition*; a Markov chain satisfying (B′) is called *reversible*.[7] (B′) is equivalent to the *self-adjointness* of P as on operator on the space $l^2(\pi)$. In this case, the spectrum of P is *real* and lies in the closed interval $[-1,1]$; we define

$$\lambda_{min} \;=\; \inf \operatorname{spec}(P \restriction \mathbf{1}^{\perp}) \tag{2.28}$$

$$\lambda_{max} \;=\; \sup \operatorname{spec}(P \restriction \mathbf{1}^{\perp}) \tag{2.29}$$

¿From (2.12) we have

$$\tau_{exp} \;=\; \frac{-1}{\log \max[|\lambda_{min}|, \lambda_{max}]} \,. \tag{2.30}$$

For many purposes, only the spectrum near $+1$ matters, so it is useful to define the *modified exponential autocorrelation time*

$$\tau'_{exp} \;=\; \begin{cases} -1/\log \lambda_{max} & \text{if } \lambda_{max} > 0 \\ +\infty & \text{if } \lambda_{max} \le 0 \end{cases} \tag{2.31}$$

Now let us apply the spectral theorem to the operator P: it follows that the autocorrelation function $\rho_{ff}(t)$ has a spectral representation

$$\rho_{ff}(t) \;=\; \int\limits_{\lambda_{min,f}}^{\lambda_{max,f}} \lambda^{|t|}\, d\sigma_{ff}(\lambda) \tag{2.32}$$

with a *nonnegative* spectral weight $d\sigma_{ff}(\lambda)$ supported on an interval $[\lambda_{min,f},\ \lambda_{max,f}] \subset [\lambda_{min},\ \lambda_{max}]$. Clearly

$$\tau_{exp,f} \;=\; \frac{-1}{\log \max[|\lambda_{min,f}|, \lambda_{max,f}]} \tag{2.33}$$

and we can define

$$\tau'_{exp,f} \;=\; \begin{cases} -1/\log \lambda_{max,f} & \text{if } \lambda_{max,f} > 0 \\ 0 & \text{if } \lambda_{max,f} \le 0 \end{cases} \tag{2.34}$$

(if $\lambda_{max,f} \ge 0$). On the other hand,

$$\tau_{int,f} \;=\; \frac{1}{2} \int\limits_{\lambda_{min,f}}^{\lambda_{max,f}} \frac{1+\lambda}{1-\lambda}\, d\sigma_{ff}(\lambda) \tag{2.35}$$

It follows that

$$\tau_{int,f} \;\le\; \frac{1}{2}\left(\frac{1+e^{-1/\tau'_{exp,f}}}{1-e^{-1/\tau'_{exp,f}}}\right) \;\le\; \frac{1}{2}\left(\frac{1+e^{-1/\tau'_{exp}}}{1-e^{-1/\tau'_{exp}}}\right) \;\approx\; \tau'_{exp} \,. \tag{2.36}$$

[7]For the physical significance of this term, see Kemeny and Snell [4, section 5.3] or Iosifescu [5, section 4.5].

Moreover, since $\lambda \mapsto (1+\lambda)/(1-\lambda)$ is a convex function, Jensen's inequality implies that

$$\tau_{int,f} \;\geq\; \frac{1}{2}\,\frac{1+\rho_{ff}(1)}{1-\rho_{ff}(1)}\;. \tag{2.37}$$

If we define the *initial autocorrelation time*

$$\tau_{init,f} \;=\; \begin{cases} -1/\log\rho_{ff}(1) & \text{if } \rho_{ff}(1) \geq 0 \\ \text{undefined} & \text{if } \rho_{ff}(1) < 0 \end{cases} \tag{2.38}$$

then these inequalities can be summarized conveniently as

$$\tau_{init,f} \;\leq\; \tilde{\tau}_{int,f} \;\leq\; \tau'_{exp,f} \;\leq\; \tau'_{exp}\;. \tag{2.39}$$

Conversely, it is not hard to see that

$$\sup_{f\in l^2(\pi)} \tau_{init,f} \;=\; \sup_{f\in l^2(\pi)} \tilde{\tau}_{int,f} \;=\; \sup_{f\in l^2(\pi)} \tau'_{exp,f} \;=\; \tau'_{exp}\;; \tag{2.40}$$

it suffices to choose f so that its spectral weight $d\sigma_{ff}$ is supported in a very small interval near λ_{max}.

Finally, let us make a remark about transition probabilities P that are "built up out of" other transition probabilities $P_1, P_2, \ldots, P_n$:

a) If $P_1, P_2, \ldots, P_n$ satisfy the stationarity condition (resp. the detailed-balance condition) for π, then so does any convex combination $P = \sum_{i=1}^n \lambda_i P_i$. Here $\lambda_i \geq 0$ and $\sum_{i=1}^n \lambda_i = 1$.

b) If $P_1, P_2, \ldots, P_n$ satisfy the stationarity condition for π, then so does the product $P = P_1 P_2 \cdots P_n$. (Note, however, that P does *not* in general satisfy the detailed-balance condition, even if the individual P_i do.[8])

Algorithmically, the convex combination amounts to choosing *randomly*, with probabilities $\{\lambda_i\}$, from among the "elementary operations" P_i. (It is crucial here that the λ_i are *constants*, independent of the current configuration of the system; only in this case does P leave π stationary in general.) Similarly, the product corresponds to performing *sequentially* the operations $P_1, P_2, \ldots, P_n$.

[8]Recall that if A and B are self-adjoint operators, then AB is self-adjoint *if and only if* A and B commute.

3 Statistical Analysis of Dynamic Monte Carlo Data

Many published Monte Carlo studies contain statements like:

> We ran for a total of 100000 iterations, discarding the first 50000 iterations (for equilibration) and then taking measurements once every 100 iterations.

It is important to emphasize that unless further information is given — namely, the autocorrelation time of the algorithm — *such statements have no value whatsoever.*

Is a run of 100000 iterations long enough? Are 50000 iterations sufficient for equilibration? That depends on how big the autocorrelation time is. The purpose of this lecture is to give some practical advice for choosing the parameters of a dynamic Monte Carlo simulation, and to give an introduction to the statistical theory that puts this advice on a sound mathematical footing.

There are two fundamental — and quite distinct — issues in dynamic Monte Carlo simulation:

- *Initialization bias.* If the Markov chain is started in a distribution α that is not equal to the stationary distribution π, then there is an "initial transient" in which the data do not reflect the desired equilibrium distribution π. This results in a *systematic error* (bias).

- *Autocorrelation in equilibrium.* The Markov chain, once it reaches equilibrium, provides *correlated* samples from π. This correlation causes the *statistical error* (variance) to be a factor $2\tau_{int,f}$ larger than in independent sampling.

Let us discuss these issues in turn.

Initialization bias. Often the Markov chain is started in some chosen configuration x; then $\alpha = \delta_x$. For example, in an Ising model, x might be the configuration with "all spins up"; this is sometimes called an *ordered* or *cold* start. Alternatively, the Markov chain might be started in a random configuration chosen according to some simple probability distribution α. For example, in an Ising model, we might initialize the spins randomly and independently, with equal probabilities of up and down; this is sometimes called a *random* or *hot* start. In all these cases, the initial distribution α is clearly *not* equal to the equilibrium distribution π. Therefore, the system is initially "out of equilibrium". Theorem 1 guarantees that the system approaches equilibrium as $t \to \infty$, but we need to know something about the *rate* of convergence to equilibrium.

Using the exponential autocorrelation time τ_{exp}, we can set an *upper bound* on the amount of time we have to wait before equilibrium is "for all practical purposes" attained. For example, if we wait a time $20\tau_{exp}$, then the deviation from equilibrium (in the l^2 sense) will be at most e^{-20} ($\approx 2 \times 10^{-9}$) times the initial deviation from equilibrium. There are two difficulties with this bound. Firstly, it is usually impossible to apply in practice, since we almost never know τ_{exp} (or a rigorous upper bound for it). Secondly, even if we can apply it, it may be overly conservative; indeed, there exist perfectly reasonable algorithms in which $\tau_{exp} = +\infty$ (see Sections 7 and 8).

Lacking rigorous knowledge of the autocorrelation time τ_{exp}, we should try to estimate it both *theoretically* and *empirically*. To make a heuristic theoretical estimate of τ_{exp}, we attempt to understand the physical mechanism(s) causing slow convergence to equilibrium; but it is always possible that we have overlooked one or more such mechanisms, and have therefore grossly underestimated τ_{exp}. To make a rough empirical estimate of τ_{exp}, we measure the autocorrelation function $C_{ff}(t)$ for a suitably large set of observables f [see below]; but there is always the danger that our chosen set of observables has failed to include one that has strong enough overlap with the slowest mode, again leading to a gross underestimate of τ_{exp}.

On the other hand, the actual rate of convergence to equilibrium from a given initial distribution α may be much faster than the worst-case estimate given by τ_{exp}. So it is usual to determine *empirically* when "equilibrium" has been achieved, by plotting selected observables as a function of time and noting when the initial transient appears to end. More sophisticated statistical tests for initialization bias can also be employed [13].

In all empirical methods of determining when "equilibrium" has been achieved, a serious danger is the possibility of *metastability*. That is, it could *appear* that equilibrium has been achieved, when in fact the system has only settled down to a long-lived (metastable) region of configuration space that may be very *far* from equilibrium. The only sure-fire protection against metastability is a *proof* of an upper bound on τ_{exp} (or more generally, on the deviation from equilibrium as a function of the elapsed time t). The next-best protection is a convincing heuristic argument that metastability is unlikely (i.e. that τ_{exp} is not too large); but as mentioned before, even if one rules out several potential physical mechanisms for metastability, it is very difficult to be certain that one has not overlooked others. If one cannot rule out metastability on theoretical grounds, then it is helpful at least to have an idea of what the possible metastable regions look like; then one can perform several runs with different initial conditions typical of each of the possible metastable regions, and test whether the answers are consistent. For example, near a first-order phase transition, most Monte Carlo methods suffer from metastability associated with transitions between configurations typical of the distinct pure phases. We can try initial conditions typical of each of these phases (e.g. for many models, a "hot" start and a "cold" start). Consistency between these runs does not *guarantee* that metastability is absent, but it does give increased confidence. Plots of observables as a function of time are also useful indicators of possible metastability.

But when all is said and done, no purely empirical estimate of τ from a run of length n can be guaranteed to be even approximately correct. What we *can* say is that if $\tau_{estimated} \ll n$, then *either* $\tau \approx \tau_{estimated}$ or else $\tau \gtrsim n$.

Once we know (or guess) the time needed to attain "equilibrium", what do we do with it? The answer is clear: we discard the data from the initial transient, up to some time n_{disc}, and include only the subsequent data in our averages. In principle, this is (asymptotically) unnecessary, because the systematic errors from this initial transient will be of order τ/n, while the statistical errors will be of order $(\tau/n)^{1/2}$. But in practice, the coefficient of τ/n in the systematic error may be fairly large, if the initial distribution is very far from equilibrium. By throwing away the data from the initial transient, we lose nothing, and avoid a potentially large systematic error.

Autocorrelation in equilibrium. As explained in the preceding lecture, the variance of the sample mean $\bar{f}$ in a dynamic Monte Carlo method is a factor $2\tau_{int,f}$ higher than it would be in independent sampling. Otherwise put, a run of length n contains only $n/2\tau_{int,f}$ "effectively independent data points".

This has several implications for Monte Carlo work. On the one hand, it means that the the *computational efficiency* of the algorithm is determined principally by its autocorrelation time. More precisely, if one wishes to compare two alternative Monte Carlo algorithms for the same problem, then *the better algorithm is the one that has the smaller autocorrelation time, when time is measured in units of computer (CPU) time.* [In general there may arise tradeoffs between "physical" autocorrelation time (i.e. τ measured in *iterations*) and computational complexity *per iteration*.] So accurate measurements of the autocorrelation time are essential to evaluating the computational efficiency of competing algorithms.

On the other hand, even for a fixed algorithm, knowledge of $\tau_{int,f}$ is essential for determining run lengths — is a run of 100000 sweeps long enough? — and for setting error bars on estimates of $\langle f \rangle$. Roughly speaking, error bars will be of order $(\tau/n)^{1/2}$; so if we want 1% accuracy, then we need a run of length $\approx 10000\tau$, and so on. Above all, there is a basic self-consistency requirement: the run length n must be $\gg$ than the estimates of τ produced by that same run, otherwise *none* of the results from that run should be believed. Of course, while self-consistency is a *necessary* condition for the trustworthiness of Monte Carlo data, it is not a *sufficient* condition; there is always the danger of metastability.

Already we can draw a conclusion about the relative importance of initialization bias and autocorrelation as difficulties in dynamic Monte Carlo work. Let us *assume* that the time for initial convergence to equilibrium is comparable to (or at least not too much larger than) the equilibrium autocorrelation time $\tau_{int,f}$ (for the observables f of interest) — this is often but not always the case. Then initialization bias is a relatively trivial problem compared to autocorrelation in equilibrium. To eliminate initialization bias, it suffices to discard $\approx 20\tau$ of the data at the beginning of the run; but to achieve a reasonably small statistical error, it is necessary to make a run of length $\approx 1000\tau$ or more. So the data that must be discarded at the beginning, n_{disc}, is a negligible fraction of the total run length n. This estimate also shows that the exact value of n_{disc} is not particularly delicate: anything between $\approx 20\tau$ and $\approx n/5$ will eliminate essentially all initialization bias while paying less than a 10% price in the final error bars.

In this remainder of this lecture I would like to discuss in more detail the statistical analysis of dynamic Monte Carlo data (assumed to be already "in equilibrium"), with emphasis on how to estimate the autocorrelation time $\tau_{int,f}$ and how to compute valid error bars. What is involved here is a branch of mathematical statistics called *time-series analysis*. An excellent exposition can be found in the books of of Priestley [14] and Anderson [15].

Let $\{f_t\}$ be a real-valued stationary stochastic process with mean

$$\mu \;\equiv\; \langle f_t \rangle \,, \tag{3.1}$$

unnormalized autocorrelation function

$$C(t) \;\equiv\; \langle f_s \, f_{s+t} \rangle \;-\; \mu^2 \,, \tag{3.2}$$

normalized autocorrelation function

$$\rho(t) \;\equiv\; C(t)/C(0) \,, \tag{3.3}$$

and integrated autocorrelation time

$$\tau_{int} \;=\; \frac{1}{2} \sum_{t\,=\,-\infty}^{\infty} \rho(t) \,. \tag{3.4}$$

Our goal is to estimate μ, $C(t)$, $\rho(t)$ and τ_{int} based on a finite (but large) sample f_1, ..., f_n from this stochastic process.

The "natural" estimator of μ is the sample mean

$$\overline{f} \;\equiv\; \frac{1}{n} \sum_{i\,=\,1}^{n} f_i \,. \tag{3.5}$$

This estimator is unbiased (i.e. $\langle \overline{f} \rangle = \mu$) and has variance

$$\mathrm{var}(\overline{f}) \;=\; \frac{1}{n} \sum_{t=-(n-1)}^{n-1} (1 - \frac{|t|}{n}) \; C(t) \tag{3.6}$$

$$\approx\; \frac{1}{n} \; (2\tau_{int}) \; C(0) \qquad \text{for } n \gg \tau \tag{3.7}$$

Thus, even if we are interested only in the static quantity μ, it is necessary to estimate the dynamic quantity τ_{int} in order to determine valid error bars for μ.

The "natural" estimator of $C(t)$ is

$$\widehat{C}(t) \;\equiv\; \frac{1}{n \;-\; |t|} \sum_{i\,=\,1}^{n\,-\,|t|} (f_i \;-\; \mu)(f_{i\,+\,|t|} \;-\; \mu) \tag{3.8}$$

if the mean μ is known, and

$$\widehat{\widehat{C}}(t) \;\equiv\; \frac{1}{n \;-\; |t|} \sum_{i\,=\,1}^{n\,-\,|t|} (f_i \;-\; \overline{f})(f_{i\,+\,|t|} \;-\; \overline{f}) \tag{3.9}$$

if the mean μ is unknown. We emphasize the conceptual distinction between the autocorrelation function $C(t)$, which for each t is a *number*, and the estimator $\widehat{C}(t)$ or $\widehat{\widehat{C}}(t)$, which for each t is a *random variable*. As will become clear, this distinction is also of *practical* importance. $\widehat{C}(t)$ is an unbiased estimator of $C(t)$, and $\widehat{\widehat{C}}(t)$ is almost unbiased (the bias is of order $1/n$) [15, p. 463]. Their variances and covariances are [15, pp. 464–471] [14, pp. 324–328]

$$\begin{aligned}
\operatorname{var}(\widehat{C}(t)) &= \frac{1}{n} \sum_{m=-\infty}^{\infty} \Big[C(m)^2 + C(m+t)C(m-t) + \kappa(t,\, m,\, m+t)\Big] \\
&\quad + o\left(\frac{1}{n}\right) &(3.10)\\
\operatorname{cov}(\widehat{C}(t), \widehat{C}(u)) &= \frac{1}{n} \sum_{m=-\infty}^{\infty} [C(m)C(m+u-t) + C(m+u)C(m-t) \\
&\quad + \kappa(t,\, m,\, m+u)] + o\left(\frac{1}{n}\right) &(3.11)
\end{aligned}$$

where $t, u \geq 0$ and κ is the connected 4-point autocorrelation function

$$\begin{aligned}
\kappa(r,\, s,\, t) &\equiv \langle (f_i - \mu)(f_{i+r} - \mu)(f_{i+s} - \mu)(f_{i+t} - \mu) \rangle \\
&\quad - C(r)C(t-s) - C(s)C(t-r) - C(t)C(s-r)\,. &(3.12)
\end{aligned}$$

To leading order in $1/n$, the behavior of $\widehat{\widehat{C}}$ is identical to that of $\widehat{C}$.

The "natural" estimator of $\rho(t)$ is

$$\widehat{\rho}(t) \;\equiv\; \widehat{C}(t)/\widehat{C}(0) \tag{3.13}$$

if the mean μ is known, and

$$\widehat{\widehat{\rho}}(t) \;\equiv\; \widehat{\widehat{C}}(t)/\widehat{\widehat{C}}(0) \tag{3.14}$$

if the mean μ is unknown. The variances and covariances of $\widehat{\rho}(t)$ and $\widehat{\widehat{\rho}}(t)$ can be computed (for large n) from (3.11); we omit the detailed formulae.

The "natural" estimator of τ_{int} would seem to be

$$\widehat{\tau}_{int} \;\stackrel{?}{\equiv}\; \frac{1}{2} \sum_{t=-(n-1)}^{n-1} \widehat{\rho}(t) \tag{3.15}$$

(or the analogous thing with $\widehat{\widehat{\rho}}$), *but this is wrong!* The estimator defined in (3.15) has a variance that does not go to zero as the sample size n goes to infinity [14, pp. 420–431], so it is clearly a very bad estimator of τ_{int}. Roughly speaking, this is because the sample autocorrelations $\widehat{\rho}(t)$ for $|t| \gg \tau$ contain much "noise" but little "signal"; and there are so many of them (order n) that the noise adds up to a total variance of order 1. (For a more detailed discussion, see [14, pp. 432–437].) The solution is to cut off the sum in (3.15) using a "window" $\lambda(t)$ which is ≈ 1 for $|t| \lesssim \tau$ but ≈ 0 for $|t| \gg \tau$:

$$\widehat{\tau}_{int} \;\equiv\; \frac{1}{2} \sum_{t=-(n-1)}^{n-1} \lambda(t)\, \widehat{\rho}(t)\;. \tag{3.16}$$

This retains most of the "signal" but discards most of the "noise". A good choice is the rectangular window

$$\lambda(t) = \begin{cases} 1 & \text{if } |t| \leq M \\ 0 & \text{if } |t| > M \end{cases} \tag{3.17}$$

where M is a suitably chosen cutoff. This cutoff introduces a bias

$$\mathrm{bias}(\widehat{\tau}_{int}) \;=\; -\frac{1}{2} \sum_{|t|>M} \rho(t) + o\left(\frac{1}{n}\right) . \tag{3.18}$$

On the other hand, the variance of $\widehat{\tau}_{int}$ can be computed from (3.11); after some algebra, one obtains

$$\mathrm{var}(\widehat{\tau}_{int}) \;\approx\; \frac{2(2M+1)}{n} \tau_{int}^2 , \tag{3.19}$$

where we have made the approximation $\tau \ll M \ll n$.[9] The choice of M is thus a tradeoff between bias and variance: the bias can be made small by taking M large enough so that $\rho(t)$ is negligible for $|t| > M$ (e.g. M = a few times τ usually suffices), while the variance is kept small by taking M to be no larger than necessary consistent with this constraint. We have found the following "automatic windowing" algorithm [16] to be convenient: choose M to be the smallest integer such that $M \geq c\,\widehat{\tau}_{int}(M)$. If $\rho(t)$ were roughly a pure exponential, then it would suffice to take $c \approx 4$ (since $e^{-4} < 2\%$). However, in many cases $\rho(t)$ is expected to have an asymptotic or pre-asymptotic decay slower than exponential, so it is usually prudent to take c at least 6, and possibly as large as 10.

We have found this automatic windowing procedure to work well in practice, *provided* that a sufficient quantity of data is available ($n \gtrsim 1000\tau$). However, at present we have very little understanding of the conditions under which this windowing algorithm may produce biased estimates of τ_{int} or of its own error bars. Further theoretical and experimental study of the windowing algorithm — e.g. experiments on various exactly-known stochastic processes, with various run lengths — would be highly desirable.

4 Conventional Monte Carlo Algorithms for Spin Models

In this lecture we describe the construction of dynamic Monte Carlo algorithms for models in statistical mechanics and quantum field theory. Recall our goal: given a probability measure π on the state space S, we wish to construct a transition matrix $P = \{p_{xy}\}$ satisfying:

[9] We have also assumed that the only strong peaks in the Fourier transform of $C(t)$ are at zero frequency. This assumption is valid if $C(t) \geq 0$, but could fail if there are strong *anti*correlations.

(A) *Irreducibility.* For each pair $x, y \in S$, there exists an $n \geq 0$ for which $p_{xy}^{(n)} > 0$.

(B) *Stationarity of* π. For each $y \in S$,

$$\sum_x \pi_x \, p_{xy} \;=\; \pi_y . \tag{4.1}$$

A sufficient condition for (B), which is often more convenient to verify, is:

(B′) *Detailed balance for* π. For each pair $x, y \in S$, $\pi_x p_{xy} = \pi_y p_{yx}$.

A very general method for constructing transition matrices satisfying detailed balance for a given distribution π was introduced in 1953 by Metropolis *et al.* [17], with a slight extension two decades later by Hastings [18]. The idea is the following: Let $P^{(0)} = \{p_{xy}^{(0)}\}$ be an *arbitrary* irreducible transition matrix on S. We call $P^{(0)}$ the *proposal matrix*; we shall use it to generate *proposed* moves $x \to y$ that will then be accepted or rejected with probabilities a_{xy} and $1 - a_{xy}$, respectively. If a proposed move is rejected, then we make a "null transition" $x \to x$. Therefore, the transition matrix $P = \{p_{xy}\}$ of the full algorithm is

$$\begin{aligned} p_{xy} &= p_{xy}^{(0)} \, a_{xy} \qquad \text{for } x \neq y \\ p_{xx} &= p_{xx}^{(0)} + \sum_{y \neq x} p_{xy}^{(0)} \, (1 - a_{xy}) \end{aligned} \tag{4.2}$$

where of course we must have $0 \leq a_{xy} \leq 1$ for all x, y. It is easy to see that P satisfies detailed balance for π if and only if

$$\frac{a_{xy}}{a_{yx}} \;=\; \frac{\pi_y \, p_{yx}^{(0)}}{\pi_x \, p_{xy}^{(0)}} \tag{4.3}$$

for all pairs $x \neq y$. But this is easily arranged: just set

$$a_{xy} \;=\; F\left(\frac{\pi_y \, p_{yx}^{(0)}}{\pi_x \, p_{xy}^{(0)}} \right) , \tag{4.4}$$

where F: $[0, +\infty] \to [0, 1]$ is any function satisfying

$$\frac{F(z)}{F(1/z)} \;=\; z \qquad \text{for all } z. \tag{4.5}$$

The choice suggested by Metropolis *et al.* is

$$F(z) \;=\; \min(z, 1) \; ; \tag{4.6}$$

this is the *maximal* function satisfying (4.5). Another choice sometimes used is

$$F(z) \;=\; \frac{z}{1+z} \, . \tag{4.7}$$

Of course, it is still necessary to check that P is irreducible; this is usually done on a case-by-case basis.

Note that if the proposal matrix $P^{(0)}$ happens to *already* satisfy detailed balance for π, then we have $\pi_y\, p^{(0)}_{yx} / \pi_x\, p^{(0)}_{xy} = 1$, so that $a_{xy} = 1$ (if we use the Metropolis choice of F) and $P = P^{(0)}$. On the other hand, no matter what $P^{(0)}$ is, we obtain a matrix P that satisfies detailed balance for π. So the Metropolis-Hastings procedure can be thought of as a prescription for minimally modifying a given transition matrix $P^{(0)}$ so that it satisfies detailed balance for π.

Many textbooks and articles describe the Metropolis-Hastings procedure only in the special case in which the proposal matrix $P^{(0)}$ is *symmetric*, namely $p^{(0)}_{xy} = p^{(0)}_{yx}$. In this case (4.4) reduces to

$$a_{xy} \;=\; F\left(\frac{\pi_y}{\pi_x}\right) \;. \tag{4.8}$$

In statistical mechanics we have

$$\pi_x \;=\; \frac{e^{-\beta E_x}}{\sum\limits_y e^{-\beta E_y}} \;=\; \frac{e^{-\beta E_x}}{Z} \;, \tag{4.9}$$

and hence

$$\frac{\pi_y}{\pi_x} \;=\; e^{-\beta(E_y - E_x)} \;. \tag{4.10}$$

Note that the partition function Z has disappeared from this expression; this is crucial, as Z is almost never explicitly computable! Using the Metropolis acceptance probability $F(z) = \min(z, 1)$, we obtain the following rules for acceptance or rejection:

- If $\Delta E \equiv E_y - E_x \le 0$, then we accept the proposal *always* (i.e. with probability 1).
- If $\Delta E > 0$, then we accept the proposal with probability $e^{-\beta\Delta E}$ (< 1). That is, we choose a random number r uniformly distributed on $[0,1]$, and we accept the proposal if $r \le e^{-\beta\Delta E}$.

But there is nothing special about $P^{(0)}$ being symmetric; *any* proposal matrix $P^{(0)}$ is perfectly legitimate, and the Metropolis-Hastings procedure is defined quite generally by (4.4).

Let us emphasize once more that the Metropolis-Hastings procedure is a *general technique*; it produces an infinite family of different algorithms depending on the choice of the proposal matrix $P^{(0)}$. In the literature the term "Metropolis algorithm" is often used to denote the algorithm resulting from some *particular* commonly-used choice of $P^{(0)}$, but it is important not to be misled.

To see the Metropolis-Hastings procedure in action, let us consider a typical statistical-mechanical model, the *Ising model*: On each site i of some finite d-dimensional lattice, we place a random variable σ_i taking the values ± 1. The Hamiltonian is

$$H(\sigma) \;=\; -\sum_{\langle ij \rangle} \sigma_i \sigma_j \;, \tag{4.11}$$

where the sum runs over all nearest-neighbor pairs. The corresponding Gibbs measure is

$$\pi(\sigma) \;=\; Z^{-1}\,\exp[-\beta H(\sigma)]\;, \tag{4.12}$$

where Z is a normalization constant (the partition function). Two different proposal matrices are in common use:

1) *Single-spin-flip (Glauber) dynamics.* Fix some site i. The proposal is to flip σ_i, hence

$$p_i^{(0)}(\{\sigma\} \to \{\sigma'\}) \;=\; \begin{cases} 1 & \text{if } \sigma_i' = -\sigma_i \text{ and } \sigma_j' = \sigma_j \text{ for all } j \neq i \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

Here $P_i^{(0)}$ is symmetric, so the acceptance probability is

$$a_i(\{\sigma\} \to \{\sigma'\}) \;=\; \min(e^{-\beta\Delta E}, 1)\;, \tag{4.14}$$

where

$$\Delta E \;\equiv\; E(\{\sigma'\}) - E(\{\sigma\}) \;=\; 2\sigma_i \sum_{j \text{ n.n. of } i} \sigma_j\;. \tag{4.15}$$

So ΔE is easily computed by comparing the status of σ_i and its neighbors.

This defines a transition matrix P_i in which only the spin at site i is touched. The full "single-spin-flip Metropolis algorithm" involves sweeping through the entire lattice in either a random or periodic fashion, i.e. either

$$P \;=\; \frac{1}{V}\sum_i P_i \qquad\qquad \text{(random site updating)} \tag{4.16}$$

or

$$P \;=\; P_{i_1} P_{i_2} \cdots P_{i_V} \qquad\qquad \text{(sequential site updating)} \tag{4.17}$$

(here V is the volume). In the former case, the transition matrix P satisfies detailed balance for π. In the latter case, P does *not* in general satisfy detailed balance for π, but it does satisfy stationarity for π, which is all that really matters.

It is easy to check that P is irreducible, except in the case of sequential site updating with $\beta = 0$.[10]

2) *Pair-interchange (Kawasaki) dynamics.* Fix a pair of sites i, j. The proposal is to interchange σ_i and σ_j, hence

$$p_{(ij)}^{(0)}(\{\sigma\} \to \{\sigma'\}) \;=\; \begin{cases} 1 & \text{if } \sigma_i' = \sigma_j,\ \sigma_j' = \sigma_i \text{ and } \sigma_k' = \sigma_k \text{ for all } k \neq i, j \\ 0 & \text{otherwise} \end{cases} \tag{4.18}$$

---

[10] **Note Added 1996:** This statement is **wrong**!!! For the Metropolis acceptance probability $F(z) = \min(z, 1)$, one may construct examples of nonergodicity for sequential site updating at *any* β [104, 105]: the idea is to construct configurations for which *every* proposed update (working sequentially) has $\Delta E = 0$ and is thus accepted. Of course, the ergodicity can be restored by taking any $F(z) < 1$, such as the choice $F(z) = z/(1 + z)$.

The rest of the formulae are analogous to those for single-spin-flip dynamics. The overall algorithm is again constructed by a random or periodic sweep over a suitable set of pairs i, j, usually taken to be nearest-neighbor pairs. It should be noted that this algorithm is *not* irreducible, as it conserves the total magnetization $\mathcal{M} = \sum_i \sigma_i$. But it is irreducible on subspaces of fixed $\mathcal{M}$ (except for sequential updating with $\beta = 0$).

A very different approach to constructing transition matrices satisfying detailed balance for π is the *heat-bath method*. This is best illustrated in a specific example. Consider again the Ising model (4.12), and focus on a single site i. Then the conditional probability distribution of σ_i, given all the other spins $\{\sigma_j\}_{j\neq i}$, is

$$P^\pi(\sigma_i \,|\, \{\sigma_j\}_{j\neq i}) \;=\; \mathrm{const}\,(\{\sigma_j\}_{j\neq i}) \;\times\; \exp\left[\beta\sigma_i \sum_{j \text{ n.n. of } i} \sigma_j\right] . \qquad (4.19)$$

(Note that this conditional distribution is precisely that of a single Ising spin σ_i in an "effective magnetic field" produced by the fixed neighboring spins σ_j.) The heat-bath algorithm updates σ_i by choosing a new spin value σ_i', *independent of the old value* σ_i, from the conditional distribution (4.19); all the other spins $\{\sigma_j\}_{j\neq i}$ remain unchanged.[11] As in the Metropolis algorithm, this operation is carried out over the whole lattice, either randomly or sequentially.

Analogous algorithms can be developed for more complicated models, e.g. $P(\varphi)$ models, σ-models and lattice gauge theories. In each case, we focus on a single field variable (holding all the other variables fixed), and give it a new value, *independent of the old value*, chosen from the appropriate conditional distribution. Of course, the feasibility of this algorithm depends on our ability to construct an efficient subroutine for generating the required single-site (or single-link) random variables. But even if this algorithm is not always the most efficient one in practice, it serves as a clear standard of comparison, which is useful in the development of more sophisticated algorithms.

A more general version of the heat-bath idea is called *partial resampling*: here we focus on a set of variables rather than only one, and the new values need not be independent of the old values. That is, we divide the variables of the system, call them $\{\varphi\}$, into two subsets, call them $\{\psi\}$ and $\{\theta\}$. For fixed values of the $\{\theta\}$ variables, π induces a conditional probability distribution of $\{\psi\}$ given $\{\theta\}$, call it $P^\pi(\{\psi\} \,|\, \{\theta\})$. Then any algorithm for updating $\{\psi\}$ with $\{\theta\}$ *fixed* that leaves invariant all of the distributions $P^\pi(\,\cdot\,|\,\{\theta\})$ will also leave invariant π. One possibility is to use an *independent resampling* of $\{\psi\}$: we throw away the old values $\{\psi\}$, and take $\{\psi'\}$ to be a new random variable chosen from the probability distribution $P^\pi(\,\cdot\,|\,\{\theta\})$, independent of the old values. Independent resampling might also called *block heat-bath updating*. On the other hand, if $\{\psi\}$ is a large set of variables, independent resampling is probably

[11]The alert reader will note that *in the Ising case* the heat-bath algorithm is equivalent to the single-spin-flip Metropolis algorithm with the choice $F(z) = z/(1+z)$ of acceptance function. But this correspondence does not hold for more complicated models.

unfeasible, but we are free to use *any* updating that leaves invariant the appropriate conditional distributions. Of course, in this generality "partial resampling" includes *all* dynamic Monte Carlo algorithms — we could just take $\{\psi\}$ to be the entire system — but it is in many cases conceptually useful to focus on some subset of variables. The partial-resampling idea will be at the heart of the multi-grid Monte Carlo method (Section 5) and the embedding algorithms (Section 6).

We have now defined a rather large class of dynamic Monte Carlo algorithms: the single-spin-flip Metropolis algorithm, the single-site heat-bath algorithm, and so on. How well do these algorithms perform? Away from phase transitions, they perform rather well. However, near a phase transition, the autocorrelation time grows rapidly. In particular, near a critical point (second-order phase transition), the autocorrelation time typically diverges as

$$\tau \sim \min(L, \xi)^z , \tag{4.20}$$

where L is the linear size of the system, ξ is the correlation length of an infinite-volume system at the same temperature, and z is a *dynamic critical exponent*. This phenomenon is called *critical slowing-down*; it severely hampers the study of critical phenomena by Monte Carlo methods. Most of the remainder of these lectures will be devoted, therefore, to describing recent progress in inventing new Monte Carlo algorithms with radically reduced critical slowing-down.

The critical slowing-down of the conventional algorithms arises fundamentally from the fact that their updates are *local*: in a single step of the algorithm, "information" is transmitted from a given site only to its nearest neighbors. Crudely one might guess that this "information" executes a random walk around the lattice. In order for the system to evolve to an "essentially new" configuration, the "information" has to travel a distance of order ξ, the (static) correlation length. One would guess, therefore, that $\tau \sim \xi^2$ near criticality, i.e. that the dynamic critical exponent z equals 2. This guess is correct for the Gaussian model (free field).[12] For other models, we have a situation analogous to theory of static critical phenomena: the dynamic critical exponent is a nontrivial number that characterizes a rather large class of algorithms (a so-called "dynamic universality class"). In any case, for most models of interest, the dynamic critical exponent for local algorithms is close to 2 (usually somewhat higher) [21]. Accurate measurements of dynamic critical exponents are, however, very difficult — even more difficult than measurements of static critical exponents — and require enormous quantities of Monte Carlo data: run lengths of $\approx 10000\tau$, when τ is itself getting large!

We can now make a rough estimate of the computer time needed to study the Ising model near its critical point, or quantum chromodynamics near the continuum limit. Each sweep of the lattice takes a time of order L^d, where d is the spatial (or space-"time") dimensionality of the model. And we need $\approx 2\tau$ sweeps in order to get *one* "effectively

---

[12] Indeed, for the Gaussian model this random-walk picture can be made rigorous: see [19] combined with [20, Section 8].

independent" sample. So this means a computer time of order $L^d\xi^z \gtrsim \xi^{d+z}$.[13] For high-precision statistics one might want 10^6 "independent" samples. The reader is invited to plug in $\xi = 100$, $d = 3$ (or $d = 4$ if you're an elementary-particle physicist) and get depressed. It should be emphasized that the factor ξ^d is inherent in *all* Monte Carlo algorithms for spin models and field theories (but not for self-avoiding walks, see Section 7). The factor ξ^z could, however, conceivably be reduced or eliminated by a more clever algorithm.

What is to be done? Our knowledge of the physics of critical slowing-down tells us that the slow modes are the long-wavelength modes, if the updating is purely local. The natural solution is therefore to speed up those modes by some sort of *collective-mode* (nonlocal) updating. It is necessary, then, to *identify physically* the appropriate collective modes, and to devise an *efficient computational algorithm* for speeding up those modes. These two goals are unfortunately in conflict; it is very difficult to devise collective-mode algorithms that are not so nonlocal that their computational cost outweighs the reduction in critical slowing-down. Specific implementations of the collective-mode idea are thus highly model-dependent. At least three such algorithms have been invented so far:

- Fourier acceleration [24]
- Multi-grid Monte Carlo (MGMC) [25, 20, 26]
- The Swendsen-Wang algorithm [27] and its generalizations

Fourier acceleration and MGMC are very similar in spirit (though quite different technically). Their performance is thus probably qualitatively similar, in the sense that they probably work well for the same models and work badly for the same models. In the next lecture we give an introduction to the MGMC method; in the following lecture we discuss algorithms of Swendsen-Wang type.

---

[13] Clearly one must take $L \gtrsim \xi$ in order to avoid severe finite-size effects. Typically one approaches the critical point with $L \approx c\xi$, where $c \approx 2-4$, and then uses finite-size scaling [22, 23] to extrapolate to the infinite-volume limit. **Note Added 1996:** Recently, radical advances have been made in applying finite-size scaling to Monte Carlo simulations (see [97, 98] and especially [99, 100, 101, 102, 103]); the preceding two sentences can now be seen to be far too pessimistic. For reliable extrapolation to the infinite-volume limit, L and ξ must both be $\gg 1$, but the ratio L/ξ can in some cases be as small as 10^{-3} or even smaller (depending on the model and on the quality of the data). However, reliable extrapolation requires careful attention to systematic errors arising from correction-to-scaling terms. In practice, reliable infinite-volume values (with both statistical and systematic errors of order a few percent) can be obtained at $\xi \sim 10^5$ from lattices of size $L \sim 10^2$, at least in some models [101, 102, 103].

5 Multi-Grid Algorithms

The phenomenon of critical slowing-down is not confined to Monte Carlo simulations: very similar difficulties were encountered long ago by numerical analysts concerned with the numerical solution of partial differential equations. An ingenious solution, now called the multi-grid (MG) method, was proposed in 1964 by the Soviet numerical analyst Fedorenko [28]: the idea is to consider, in addition to the original ("fine-grid") problem, a sequence of auxiliary "coarse-grid" problems that approximate the behavior of the original problem for excitations at successively longer length scales (a sort of "coarse-graining" procedure). The local updates of the traditional algorithms are then supplemented by coarse-grid updates. To a present-day physicist, this philosophy is remarkably reminiscent of the renormalization group — so it is all the more remarkable that it was invented two years before the work of Kadanoff [29] and seven years before the work of Wilson [30]! After a decade of dormancy, multi-grid was revived in the mid-1970's [31], and was shown to be an extremely efficient computational method. In the 1980's, multi-grid methods have become an active area of research in numerical analysis, and have been applied to a wide variety of problems in *classical* physics [32, 33]. Very recently [25] it was shown how a stochastic generalization of the multi-grid method — multi-grid Monte Carlo (MGMC) — can be applied to problems in *statistical*, and hence also Euclidean *quantum*, physics.

In this lecture we begin by giving a brief introduction to the deterministic multi-grid method; we then explain the stochastic analogue.[14] But it is worth indicating now the basic idea behind this generalization. There is a strong analogy between solving lattice systems of equations (such as the discrete Laplace equation) and making Monte Carlo simulations of lattice random fields. Indeed, given a Hamiltonian $H(\varphi)$, the deterministic problem is that of *minimizing* $H(\varphi)$, while the stochastic problem is that of *generating random samples* from the Boltzmann-Gibbs probability distribution $e^{-\beta H(\varphi)}$. The statistical-mechanical problem reduces to the deterministic one in the zero-temperature limit $\beta \to +\infty$.

Many (but not all) of the deterministic iterative algorithms for minimizing $H(\varphi)$ can be generalized to stochastic iterative algorithms — that is, dynamic Monte Carlo methods — for generating random samples from $e^{-\beta H(\varphi)}$. For example, the Gauss-Seidel algorithm for minimizing H and the heat-bath algorithm for random sampling from $e^{-\beta H}$ are very closely related. Both algorithms sweep successively through the lattice, working on one site x at a time. The Gauss-Seidel algorithm updates φ_x so as to *minimize* the Hamiltonian $H(\varphi) = H(\varphi_x, \{\varphi_y\}_{y \neq x})$ when all the other fields $\{\varphi_y\}_{y \neq x}$ are held fixed at their current values. The heat-bath algorithm gives φ_x a new *random* value chosen from the probability distribution $\exp[-H(\varphi_x, \{\varphi_y\}_{y \neq x})]$, with all the fields $\{\varphi_y\}_{y \neq x}$ again held fixed. As $\beta \to +\infty$ the heat-bath algorithm approaches Gauss-Seidel. A similar correspondence holds between MG and MGMC.

[14] For an excellent introduction to the deterministic multi-grid method, see Briggs [34]; more advanced topics are covered in the book of Hackbusch [32]. Both MG and MGMC are discussed in detail in [20].

Before entering into details, let us emphasize that although the multi-grid method and the block-spin renormalization group (RG) are based on very similar *philosophies* — dealing with a single length scale at a time — they are in fact *very different*. In particular, the conditional coarse-grid Hamiltonian employed in the MGMC method is *not* the same as the renormalized Hamiltonian given by a block-spin RG transformation. The RG transformation computes the *marginal*, not the conditional, distribution of the block means — that is, it *integrates* over the complementary degrees of freedom, while the MGMC method *fixes* these degrees of freedom at their current (random) values. The conditional Hamiltonian employed in MGMC is given by an explicit finite expression, while the marginal (RG) Hamiltonian cannot be computed in closed form. The failure to appreciate these distinctions has unfortunately led to much confusion in the literature.[15]

5.1 Deterministic Multi-Grid

In this section we give a pedagogical introduction to multi-grid methods in the simplest case, namely the solution of deterministic linear or nonlinear systems of equations.

Consider, for purposes of exposition, the lattice Poisson equation $-\Delta\varphi = f$ in a region $\Omega \subset \mathbf{Z}^d$ with zero Dirichlet data. Thus, the equation is

$$(-\Delta\varphi)_x \equiv 2d\varphi_x - \sum_{x' \colon |x-x'|=1} \varphi_{x'} \;=\; f_x \tag{5.1}$$

for $x \in \Omega$, with $\varphi_x \equiv 0$ for $x \notin \Omega$. This problem is equivalent to minimizing the quadratic Hamiltonian

$$H(\varphi) \;=\; \frac{1}{2}\sum_{\langle xy\rangle}(\varphi_x - \varphi_y)^2 - \sum_x f_x\varphi_x \;. \tag{5.2}$$

More generally, we may wish to solve a linear system

$$A\varphi \;=\; f \;, \tag{5.3}$$

where for simplicity we shall assume A to be symmetric and positive-definite. This problem is equivalent to minimizing

$$H(\varphi) \;=\; \frac{1}{2}\langle\varphi, A\varphi\rangle - \langle f, \varphi\rangle \;. \tag{5.4}$$

Later we shall consider also non-quadratic Hamiltonians.

Our goal is to devise a rapidly convergent iterative method for solving numerically the linear system (5.3). We shall restrict attention to first-order stationary linear iterations of the general form

$$\varphi^{(n+1)} \;=\; M\varphi^{(n)} + Nf \;, \tag{5.5}$$

[15] For further discussion, see [20, Section 10.1].

where $\varphi^{(0)}$ is an arbitrary initial guess for the solution. Obviously, we must demand at the very least that the true solution $\varphi \equiv A^{-1} f$ be a fixed point of (5.5); imposing this condition for all f, we conclude that

$$N \; = \; (I - M) A^{-1} \; . \tag{5.6}$$

The iteration (5.5) is thus completely specified by its *iteration matrix* M. Moreover, (5.5)-(5.6) imply that the error $e^{(n)} \equiv \varphi^{(n)} - \varphi$ satisfies

$$e^{(n+1)} \; = \; M e^{(n)} \; . \tag{5.7}$$

That is, the iteration matrix is the amplification matrix for the error. It follows easily that the iteration (5.5) is convergent for all initial vectors $\varphi^{(0)}$ if and only if the spectral radius $\rho(M) \equiv \lim_{n\to\infty} ||M^n||^{1/n}$ is < 1; and in this case the convergence is exponential with asymptotic rate at least $\rho(M)$, i.e.

$$||\varphi^{(n)} - \varphi|| \; \leq \; K n^p \rho(M)^n \tag{5.8}$$

for some $K, p < \infty$ (K depends on $\varphi^{(0)}$).

Now let us return to the specific system (5.1). One simple iterative algorithm arises by solving (5.1) repeatedly for φ_x:

$$\varphi_x^{(n+1)} \; = \; \frac{1}{2d} \left[\sum_{x' :\, |x - x'| = 1} \varphi_{x'}^{(n)} + f_x \right] . \tag{5.9}$$

(5.9) is called the *Jacobi iteration*. It is convenient to consider also a slight generalization of (5.9): let $0 < \omega \; \leq \; 1$, and define

$$\varphi_x^{(n+1)} \; = \; (1 - \omega) \varphi_x^{(n)} + \frac{\omega}{2d} \left[\sum_{x' :\, |x - x'| = 1} \varphi_{x'}^{(n)} + f_x \right] . \tag{5.10}$$

(5.10) is called the *damped Jacobi iteration* with damping parameter ω; for $\omega = 1$ it reduces to the ordinary Jacobi iteration.

It can be shown [35] that the spectral radius $\rho(M_{DJ,\omega})$ of the damped Jacobi iteration matrix is less than 1, so that the iteration (5.10) converges exponentially to the solution φ. This would appear to be a happy situation. Unfortunately, however, the convergence factor $\rho(M_{DJ,\omega})$ is usually very close to 1, so that many iterations are required in order to reduce the error $||\varphi^{(n)} - \varphi||$ to a small fraction of its initial value. Insight into this phenomenon can be gained by considering the simple *model problem* in which the domain Ω is a square $\{1, \ldots, L\} \times \{1, \ldots, L\}$. In this case we can solve exactly for the eigenvectors and eigenvalues of $M_{DJ,\omega}$: they are

$$\varphi_x^{(p)} \; = \; \sin p_1 x_1 \sin p_2 x_2 \tag{5.11}$$

$$\lambda_p \; = \; (1 - \omega) + \frac{\omega}{2} (\cos p_1 + \cos p_2) \tag{5.12}$$

where $p_1, p_2 = \frac{\pi}{L+1}, \frac{2\pi}{L+1}, \ldots, \frac{L\pi}{L+1}$. The spectral radius of $M_{DJ,\omega}$ is the eigenvalue of largest magnitude, namely

$$\begin{aligned}\rho(M_{DJ,\omega}) \;=\; \lambda_{\frac{\pi}{L+1},\frac{\pi}{L+1}} &\;=\; 1-\omega\left[1-\cos\frac{\pi}{L+1}\right] \\ &\;=\; 1-O(L^{-2})\,. \end{aligned} \tag{5.13}$$

It follows that $O(L^2)$ iterations are needed for the damped Jacobi iteration to converge adequately. This represents an enormous computational labor when L is large.

It is easy to see what is going on here: *the slow modes* ($\lambda_p \approx 1$) *are the long-wavelength modes* ($p_1, p_2 \ll 1$). [If $\omega \approx 1$, then some modes with wavenumber $p = (p_1, p_2) \approx (\pi, \pi)$ have eigenvalue $\lambda_p \approx -1$ and so also are slow. This phenomenon can be avoided by taking ω significantly less than 1; for simplicity we shall henceforth take $\omega = \frac{1}{2}$, which makes $\lambda_p \geq 0$ for all p.] It is also easy to see physically *why* the long-wavelength modes are slow. The key fact is that the (damped) Jacobi iteration is *local*: in a single step of the algorithm, "information" is transmitted only to nearest neighbors. One might guess that this "information" executes a random walk around the lattice; and for the true solution to be reached, "information" must propagate from the boundaries to the interior (and back and forth until "equilibrium" is attained). This takes a time of order L^2, in agreement with (5.13). In fact, this random-walk picture can be made rigorous [19].

This is an example of a *critical phenomenon*, in precisely the same sense that the term is used in statistical mechanics. The Laplace operator $A = -\Delta$ is critical, inasmuch as its Green function A^{-1} has long-range correlations (power-law decay in dimension $d > 2$, or growth in $d \leq 2$). This means that the solution of Poisson's equation in one region of the lattice depends strongly on the solution in distant regions of the lattice; "information" must propagate globally in order for "equilibrium" to be reached. Put another way, excitations at many length scales are significant, from one lattice spacing at the smallest to the entire lattice at the largest. The situation would be very different if we were to consider instead the Helmholtz-Yukawa equation $(-\Delta + m^2)\varphi = f$ with $m > 0$: its Green function has exponential decay with characteristic length m^{-1}, so that regions of the lattice separated by distances $\gg m^{-1}$ are essentially decoupled. In this case, "information" need only propagate a distance of order $\min(m^{-1}, L)$ in order for "equilibrium" to be reached. This takes a time of order $\min(m^{-2}, L^2)$, an estimate which can be confirmed rigorously by computing the obvious generalization of (5.12)–(5.13). On the other hand, as $m \to 0$ we recover the Laplace operator with its attendant difficulties: $m = 0$ is a *critical point*. We have here an example of *critical slowing-down* in classical physics.

The general structure of a remedy should now be obvious to physicists reared on the renormalization group: don't try to deal with all length scales at once, but define instead a *sequence* of problems in which each length scale, beginning with the smallest and working towards the largest, can be dealt with separately. An algorithm of precisely this form was proposed in 1964 by the Soviet numerical analyst Fedorenko [28], and is now called the *multi-grid method*.

Note first that the *only* slow modes in the damped Jacobi iteration are the long-wavelength modes (provided that ω is not near 1): as long as, say, $\max(p_1, p_2) \geq \frac{\pi}{2}$, we have $0 \leq \lambda_p \leq \frac{3}{4}$ (for $\omega = \frac{1}{2}$), independent of L. It follows that the short-wavelength components of the error $e^{(n)} \equiv \varphi^{(n)} - \varphi$ can be effectively killed by a few (say, five or ten) damped Jacobi iterations. The remaining error has primarily long-wavelength components, and so is slowly varying in x-space. But a slowly varying function can be well represented on a coarser grid: if, for example, we were told $e_x^{(n)}$ only at *even* values of x, we could nevertheless reconstruct with high accuracy the function $e_x^{(n)}$ at *all* x by, say, linear interpolation. This suggests an improved algorithm for solving (5.1): perform a few damped Jacobi iterations on the original grid, until the (unknown) error is smooth in x-space; then set up an auxiliary coarse-grid problem whose solution will be approximately this error (this problem will turn out to be a Poisson equation on the coarser grid); perform a few damped Jacobi iterations on the coarser grid; and then transfer (interpolate) the result back to the original (fine) grid and add it in to the current approximate solution.

There are two advantages to performing the damped Jacobi iterations on the coarse grid. Firstly, the iterations take less work, because there are fewer lattice points on the coarse grid (2^{-d} times as many for a factor-of-2 coarsening in d dimensions). Secondly, with respect to the coarse grid the long-wavelength modes no longer have such long wavelength: *their wavelength has been halved* (i.e. their wavenumber has been doubled). This suggests that those modes with, say, $\max(p_1, p_2) \geq \frac{\pi}{4}$ can be effectively killed by a few damped Jacobi iterations on the coarse grid. And then we can transfer the remaining (smooth) error to a yet coarser grid, and so on recursively. These are the essential ideas of the multi-grid method.

Let us now give a precise definition of the multi-grid algorithm. For simplicity we shall restrict attention to problems defined in variational form[16]: thus, the goal is to minimize a real-valued function ("Hamiltonian") $H(\varphi)$, where φ runs over some N-dimensional real vector space U. We shall treat quadratic and non-quadratic Hamiltonians on an equal footing. In order to specify the algorithm we must specify the following ingredients:

1) A sequence of *coarse-grid spaces* $U_M \equiv U$, U_{M-1}, U_{M-2}, $\ldots$, U_0. Here $\dim U_l \equiv N_l$ and $N = N_M > N_{M-1} > N_{M-2} > \cdots > N_0$.

2) *Prolongation* (or "interpolation") *operators* $p_{l,l-1}$: $U_{l-1} \to U_l$ for $1 \leq l \leq M$.

3) *Basic* (or "smoothing") *iterations* $\mathcal{S}_l$: $U_l \times \mathcal{H}_l \to U_l$ for $0 \leq l \leq M$. Here $\mathcal{H}_l$ is a space of "possible Hamiltonians" defined on U_l; we discuss this in more detail below. The role of $\mathcal{S}_l$ is to take an approximate minimizer φ'_l of the Hamiltonian H_l and compute a new (hopefully better) approximate minimizer $\varphi''_l = \mathcal{S}_l(\varphi'_l, H_l)$. [For

[16] In fact, the multi-grid method can be applied to the solution of linear or nonlinear systems of equations, whether or not these equations come from a variational principle. See, for example, [32] and [20, Section 2].

the present we can imagine that $\mathcal{S}_l$ consists of a few iterations of damped Jacobi for the Hamiltonian H_l.] Most generally, we shall use two smoothing iterations, $\mathcal{S}_l^{pre}$ and $\mathcal{S}_l^{post}$; they may be the same, but need not be.

4) *Cycle control parameters* (integers) $\gamma_l \geq 1$ for $1 \leq l \leq M$, which control the number of times that the coarse grids are visited.

We discuss these ingredients in more detail below.

The multi-grid algorithm is then defined recursively as follows:

procedure $mgm(l, \varphi, H_l)$

comment This algorithm takes an approximate minimizer φ of the Hamiltonian H_l, and overwrites it with a better approximate minimizer.

$\varphi \leftarrow \mathcal{S}_l^{pre}(\varphi, H_l)$

if $l > 0$ **then**

  compute $H_{l-1}(\,\cdot\,) \equiv H_l(\varphi + p_{l,l-1}\,\cdot\,)$

  $\psi \leftarrow 0$

  for $j = 1$ **until** γ_l **do** $mgm(l-1, \psi, H_{l-1})$

  $\varphi \leftarrow \varphi + p_{l,l-1}\psi$

endif

$\varphi \leftarrow \mathcal{S}_l^{post}(\varphi, H_l)$

end

Here is what is going on: We wish to solve the minimize the Hamiltonian H_l, and are given as input an approximate minimizer φ. The algorithm consists of three steps:

1) *Pre-smoothing.* We apply the basic iteration (e.g. a few sweeps of damped Jacobi) to the given approximate minimizer. This produces a better approximate minimizer in which the high-frequency (short-wavelength) components of the error have been reduced significantly. Therefore, the error, although still large, is *smooth* in x-space (whence the name "smoothing iteration").

2) *Coarse-grid correction.* We want to move rapidly towards the minimizer φ^* of H_l, using coarse-grid updates. Because of the pre-smoothing, the error $\varphi - \varphi^*$ is a smooth function in x-space, so it should be well approximated by fields in the range of the prolongation operator $p_{l,l-1}$. We will therefore carry out a coarse-grid update in which φ is replaced by $\varphi + p_{l,l-1}\psi$, where ψ lies in the coarse-grid subspace U_{l-1}. A sensible goal is to attempt to choose ψ so as to minimize H_l; that is, we attempt to minimize

$$H_{l-1}(\psi) \;\equiv\; H_l(\varphi + p_{l,l-1}\psi)\;. \tag{5.14}$$

To carry out this approximate minimization, we use a few (γ_l) iterations of the best algorithm we know — namely, multi-grid itself! And we start at the best approximate minimizer we know, namely $\psi = 0$! The goal of this coarse-grid correction step is to reduce significantly the low-frequency components of the error in φ (hopefully without creating large new high-frequency error components).

3) *Post-smoothing.* We apply, for good measure, a few more sweeps of the basic smoother. (This would protect against any high-frequency error components which may inadvertently have been created by the coarse-grid correction step.)

The foregoing constitutes, of course, a single step of the multi-grid algorithm. In practice this step would be repeated several times, as in any other iteration, until the error has been reduced to an acceptably small value. The advantage of multi-grid over the traditional (e.g. damped Jacobi) iterative methods is that, with a suitable choice of the ingredients $p_{l,l-1}$, $\mathcal{S}_l$ and so on, only a few (maybe five or ten) iterations are needed to reduce the error to a small value, *independent of the lattice size L.* This contrasts favorably with the behavior (5.13) of the damped Jacobi method, in which $O(L^2)$ iterations are needed.

The multi-grid algorithm is thus a general framework; the user has considerable freedom in choosing the specific ingredients, which must be adapted to the specific problem. We now discuss briefly each of these ingredients; more details can be found in Chapter 3 of the book of Hackbusch [32].

Coarse grids. Most commonly one uses a uniform factor-of-2 coarsening between each grid Ω_l and the next coarser grid Ω_{l-1}. The coarse-grid points could be either a subset of the fine-grid points (Fig. 1) or a subset of the dual lattice (Fig. 2). These schemes have obvious generalizations to higher-dimensional cubic lattices. In dimension $d = 2$, another possibility is a uniform factor-of-$\sqrt{2}$ coarsening (Fig. 3); note that the coarse grid is again a square lattice, rotated by 45°. Figs. 1–3 are often referred to as "standard coarsening", "staggered coarsening", and "red-black (or checkerboard) coarsening", respectively. Coarsenings by a larger factor (e.g. 3) could also be considered, but are generally disadvantageous. Note that each of the above schemes works also for periodic boundary conditions provided that the linear size L_l of the grid Ω_l is *even.* For this reason it is most convenient to take the linear size $L \equiv L_M$ of the original (finest) grid $\Omega \equiv \Omega_M$ to be a power of 2, or at least a power of 2 times a small integer. Other definitions of coarse grids (e.g. anisotropic coarsening) are occasionally appropriate.

Prolongation operators. For a coarse grid as in Fig. 2, a natural choice of prolongation operator is *piecewise-constant injection*:

$$(p_{l,l-1}\varphi_{l-1})_{x_1\pm\frac{1}{2},x_2\pm\frac{1}{2}} \;=\; (\varphi_{l-1})_{x_1,x_2} \qquad \text{for all } x \in \Omega_{l-1} \tag{5.15}$$

(illustrated here for $d = 2$). It can be represented in an obvious shorthand notation by the stencil

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} . \tag{5.16}$$

For a coarse grid as in Fig. 1, a natural choice is *piecewise-linear interpolation*, one

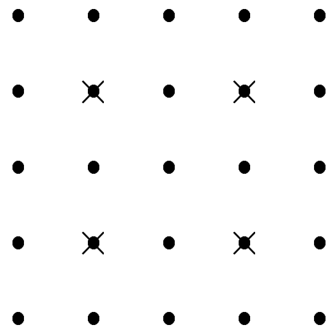

Figure 1: Standard coarsening (factor-of-2) in dimension $d = 2$. Dots are fine-grid sites and crosses are coarse-grid sites.

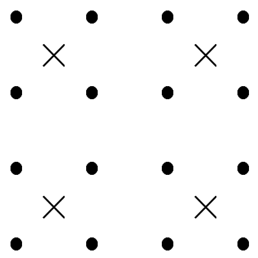

Figure 2: Staggered coarsening (factor-of-2) in dimension $d = 2$.

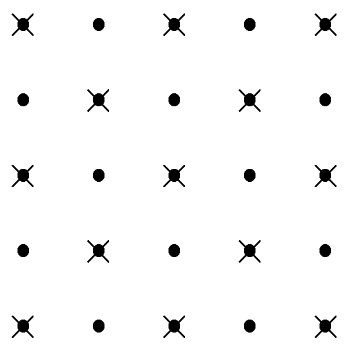

Figure 3: Red-black coarsening (factor-of-$\sqrt{2}$) in dimension $d = 2$.

example of which is the *nine-point prolongation*

$$\begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} . \tag{5.17}$$

Higher-order interpolations (e.g. quadratic or cubic) can also be considered. All these prolongation operators can easily be generalized to higher dimensions.

We have ignored here some important subtleties concerning the treatment of the boundaries in defining the prolongation operator. Fortunately we shall not have to worry much about this problem, since most applications in statistical mechanics and quantum field theory use periodic boundary conditions.

Coarse-grid Hamiltonians. What does the coarse-grid Hamiltonian H_{l-1} look like? If the fine-grid Hamiltonian H_l is quadratic,

$$H_l(\varphi) \;=\; \frac{1}{2}\langle \varphi, A_l \varphi\rangle - \langle f_l, \varphi\rangle \;, \tag{5.18}$$

then so is the coarse-grid Hamiltonian H_{l-1}:

$$\begin{aligned} H_{l-1}(\psi) &\equiv H_l(\varphi + p_{l,l-1}\psi) & (5.19)\\ &= \frac{1}{2}\langle \psi, A_{l-1}\psi\rangle - \langle d, \psi\rangle + \text{const} \;, & (5.20) \end{aligned}$$

where

$$\begin{aligned} A_{l-1} &\equiv p^*_{l,l-1}\, A_l\, p_{l,l-1} & (5.21)\\ d &\equiv p^*_{l,l-1}(f - A_l\varphi) & (5.22) \end{aligned}$$

The coarse-grid problem is thus also a linear equation whose right-hand side is just the "coarse-graining" of the residual $r \equiv f - A_l\varphi$; this coarse-graining is performed using the adjoint of the interpolation operator $p_{l,l-1}$. The exact form of the coarse-grid operator A_{l-1} depends on the fine-grid operator A_l and on the choice of interpolation operator $p_{l,l-1}$. For example, if A_l is the nearest-neighbor Laplacian and $p_{l,l-1}$ is piecewise-constant injection, then it is easily checked that A_{l-1} is also a nearest-neighbor Laplacian (multiplied by an extra 2^{d-1}). On the other hand, if $p_{l,l-1}$ is piecewise-linear interpolation, then A_{l-1} will have nearest-neighbor *and next-nearest-neighbor* terms (but nothing worse than that).

Clearly, the point is to choose classes of Hamiltonians $\mathcal{H}_l$ with the property that if $H_l \in \mathcal{H}_l$ and $\varphi \in U_l$, then the coarse-grid Hamiltonian H_{l-1} defined by (5.14) necessarily lies in $\mathcal{H}_{l-1}$. In particular, it is convenient (though not in principle necessary) to choose all the Hamiltonians to have the same "functional form"; this functional form must be one which is stable under the coarsening operation (5.14). For example, suppose that the Hamiltonian H_l is a φ^4 theory with nearest-neighbor gradient term and possibly site-dependent coefficients:

$$H_l(\varphi) \;=\; \frac{\alpha}{2} \sum_{|x-x'|=1} (\varphi_x - \varphi_{x'})^2 + \sum_x V_x(\varphi_x), \tag{5.23}$$

where

$$V_x(\varphi_x) \;=\; \lambda\varphi_x^4 + \kappa_x\varphi_x^3 + A_x\varphi_x^2 + h_x\varphi_x. \tag{5.24}$$

Suppose, further, that the prolongation operator $p_{l,l-1}$ is piecewise-constant injection (5.15). Then the coarse-grid Hamiltonian $H_{l-1}(\psi) \equiv H_l(\varphi + p_{l,l-1}\psi)$ can easily be computed: it is

$$H_{l-1}(\psi) \;=\; \frac{\alpha'}{2} \sum_{|y-y'|=1} (\psi_y - \psi_{y'})^2 + \sum_y V'_y(\psi_y) + \text{const}, \tag{5.25}$$

where

$$V'_y(\psi_y) \;=\; \lambda'\psi_y^4 + \kappa'_y\psi_y^3 + A'_y\psi_y^2 + h'_y\psi_y \tag{5.26}$$

and

$$\alpha' \;=\; 2^{d-1}\alpha \tag{5.27}$$

$$\lambda' \;=\; 2^d\lambda \tag{5.28}$$

$$\kappa'_y \;=\; \sum_{x\in B_y} (4\lambda\varphi_x + \kappa_x) \tag{5.29}$$

$$A'_y \;=\; \sum_{x\in B_y} (6\lambda\varphi_x^2 + 3\kappa_x\varphi_x + A_x) \tag{5.30}$$

$$h'_y \;=\; \sum_{x\in B_y} (4\lambda\varphi_x^3 + 3\kappa_x\varphi_x^2 + 2A_x\varphi_x + h_x) \tag{5.31}$$

Here B_y is the block consisting of those 2^d sites of grid Ω_l which are affected by interpolation from the coarse-grid site $y \in \Omega_{l-1}$ (see Figure 2). Note that the coarse-grid

Hamiltonian H_{l-1} has the same functional form as the "fine-grid" Hamiltonian H_l: it is specified by the coefficients α', λ', κ'_y, A'_y and h'_y. The step "*compute* H_{l-1}" therefore means to compute these coefficients. Note also the importance of allowing in (5.23) for φ^3 and φ terms and for site-dependent coefficients: even if these are not present in the original Hamiltonian $H \equiv H_M$, they will be generated on coarser grids. Finally, we emphasize that the coarse-grid Hamiltonian H_{l-1} depends implicitly on the current value of the fine-lattice field $\varphi \in U_l$; although our notation suppresses this dependence, it should be kept in mind.

Basic (smoothing) iterations. We have already discussed the damped Jacobi iteration as one possible smoother. Note that in this method only the "old" values $\varphi^{(n)}$ are used on the right-hand side of (5.9)/(5.10), even though for some of the terms the "new" value $\varphi^{(n+1)}$ may already have been computed. An alternative algorithm is to use at each stage on the right-hand side the "newest" available value. This algorithm is called the *Gauss-Seidel iteration.*[17] Note that the Gauss-Seidel algorithm, unlike the Jacobi algorithm, depends on the *ordering* of the grid points. For example, if a 2-dimensional grid is swept in *lexicographic order* $(1,1)$, $(2,1)$, $\ldots$, $(L,1)$, $(1,2)$, $(2,2)$, $\ldots$, $(L,2)$, $\ldots$, $(1,L)$, $(2,L)$, $\ldots$, (L,L), then the Gauss-Seidel iteration becomes

$$\varphi^{(n+1)}_{x_1,x_2} \;=\; \frac{1}{4}[\varphi^{(n)}_{x_1+1,x_2} + \varphi^{(n+1)}_{x_1-1,x_2} + \varphi^{(n)}_{x_1,x_2+1} + \varphi^{(n+1)}_{x_1,x_2-1} + f_{x_1,x_2}]. \tag{5.32}$$

Another convenient ordering is the *red-black* (or *checkerboard*) *ordering*, in which the "red" sublattice $\Omega^r = \{x \in \Omega\colon\ x_1 + \cdots + x_d \text{ is even}\}$ is swept first, followed by the "black" sublattice $\Omega^b = \{x \in \Omega\colon\ x_1 + \cdots + x_d \text{ is odd}\}$. Note that the ordering of the grid points *within* each sublattice is irrelevant [for the usual nearest-neighbor Laplacian (5.1)], since the matrix A does not couple sites of the same color. This means that red-black Gauss-Seidel is particularly well suited to vector or parallel computation. Note that the red-black ordering makes sense with periodic boundary conditions only if the linear size L_l of the grid Ω_l is *even.*

It turns out that Gauss-Seidel is a better smoother than damped Jacobi (even if the latter is given its optimal ω). Moreover, Gauss-Seidel is easier to program and requires only half the storage space (no need for separate storage of "old" and "new" values). The only reason we introduced damped Jacobi at all is that it is easier to understand and to analyze.

Many other smoothing iterations can be considered, and can be advantageous in anisotropic or otherwise singular problems [32, Section 3.3 and Chapters 10–11]. But we shall stick to ordinary Gauss-Seidel, usually with red-black ordering.

Thus, $\mathcal{S}_l^{pre}$ and $\mathcal{S}_l^{post}$ will consist, respectively, of m_1 and m_2 iterations of the Gauss-Seidel algorithm. The balance between pre-smoothing and post-smoothing is usually not very crucial; only the total $m_1 + m_2$ seems to matter much. Indeed, one (but not

[17] It is amusing to note that "Gauss did not use a cyclic order of relaxation, and ... Seidel specifically recommended against using it" [36, p. 44n]. See also [37].

both!) of m_1 or m_2 could be zero, i.e. either the pre-smoothing or the post-smoothing could be omitted entirely. Increasing m_1 and m_2 improves the convergence rate of the multi-grid iteration, but at the expense of increased computational labor per iteration. The optimal tradeoff seems to be achieved in most cases with $m_1 + m_2$ between about 2 and 4. The coarsest grid Ω_0 is a special case: it usually has so few grid points (perhaps only one!) that $\mathcal{S}_0$ can be an exact solver.

The variational point of view gives special insight into the Gauss-Seidel algorithm, and shows how to generalize it to non-quadratic Hamiltonians. When updating site x, the new value φ'_x is chosen so as to *minimize* the Hamiltonian $H(\varphi) = H(\varphi_x, \{\varphi_y\}_{y \neq x})$ when all the other fields $\{\varphi_y\}_{y\neq x}$ are held fixed at their current values. The natural generalization of Gauss-Seidel to non-quadratic Hamiltonians is to adopt this variational definition: φ'_x should be the *absolute minimizer* of $H(\varphi) = H(\varphi_x, \{\varphi_y\}_{y\neq x})$. [If the absolute minimizer is non-unique, then one such minimizer is chosen by some arbitrary rule.] This algorithm is called nonlinear Gauss-Seidel with exact minimization (NL-GSEM) [38]. This definition of the algorithm presupposes, of course, that it is feasible to carry out the requisite exact one-dimensional minimizations. For example, for a φ^4 theory it would be necessary to compute the absolute minimum of a quartic polynomial in one variable. In practice these one-dimensional minimizations might themselves be carried out iteratively, e.g. by some variant of Newton's method.

Cycle control parameters and computational labor. Usually the parameters γ_l are all taken to be equal, i.e. $\gamma_l = \gamma \geq 1$ for $1 \leq l \leq M$. Then one iteration of the multi-grid algorithm at level M comprises one visit to grid M, γ visits to grid $M-1$, γ^2 visits to grid $M-2$, and so on. Thus, γ determines the degree of emphasis placed on the coarse-grid updates. ($\gamma = 0$ would correspond to the pure Gauss-Seidel iteration on the finest grid alone.)

We can now estimate the computational labor required for one iteration of the multi-grid algorithm. Each visit to a given grid involves $m_1 + m_2$ Gauss-Seidel sweeps on that grid, plus some computation of the coarse-grid Hamiltonian and the prolongation. The work involved is proportional to the number of lattice points on that grid. Let W_l be the work required for these operations on grid l. Then, for grids defined by a factor-of-2 coarsening in d dimensions, we have

$$W_l \;\approx\; 2^{-d(M-l)} W_M \,, \tag{5.33}$$

so that the total work for one multi-grid iteration is

$$\begin{aligned} work(MG) &= \sum_{l=M}^{0} \gamma^{M-l} W_l \\ &\approx W_M \sum_{l=M}^{0} (\gamma 2^{-d})^{M-l} \\ &\leq W_M (1-\gamma 2^{-d})^{-1} \qquad \text{if } \gamma < 2^d \,. \end{aligned} \tag{5.34}$$

Thus, provided that $\gamma < 2^d$, the work required for one entire multi-grid iteration is no more than $(1-\gamma 2^{-d})^{-1}$ times the work required for $m_1 + m_2$ Gauss-Seidel iterations (plus a little auxiliary computation) on the finest grid alone — *irrespective of the total number of levels.* The most common choices are $\gamma = 1$ (which is called the V-cycle) and $\gamma = 2$ (the W-cycle).

Convergence proofs. For certain classes of Hamiltonians H — primarily quadratic ones — and suitable choices of the coarse grids, prolongations, smoothing iterations and cycle control parameters, it can be proven rigorously[18] that the multi-grid iteration matrices M_l satisfy a *uniform* bound

$$||M_l|| \;\leq\; C \;<\; 1 \,, \tag{5.35}$$

valid *irrespective of the total number of levels.* Thus, a fixed number of multi-grid iterations (maybe five or ten) are sufficient to reduce the error to a small value, *independent of the lattice size L.* In other words, *critical slowing-down has been completely eliminated.*

The rigorous convergence proofs are somewhat arcane, so we cannot describe them here in any detail, but certain general features are worth noting. The convergence proofs are most straightforward when linear or higher-order interpolation and restriction are used, and $\gamma > 1$ (e.g. the W-cycle). When either low-order interpolation (e.g. piecewise-constant) or $\gamma = 1$ (the V-cycle) is used, the convergence proofs become much more delicate. Indeed, if *both* piecewise-constant interpolation and a V-cycle are used, then the uniform bound (5.35) has *not* yet been proven, and it is most likely *false*! To some extent these features may be artifacts of the current methods of proof, but we suspect that they do also reflect real properties of the multi-grid method, and so the convergence proofs may serve as guidance for practice. For example, in our work we have used piecewise-constant interpolation (so as to preserve the simple nearest-neighbor coupling on the coarse grids), and thus for safety we stick to the W-cycle. There is in any case much room for further research, both theoretical and experimental.

To recapitulate, the extraordinary efficiency of the multi-grid method arises from the combination of two key features:

1) The convergence estimate (5.35). This means that only $O(1)$ iterations are needed, independent of the lattice size L.

2) The work estimate (5.34). This means that each iteration requires only a computational labor of order L^d (the fine-grid lattice volume).

It follows that the complete solution of the minimization problem, to any specified accuracy ε, requires a computational labor of order L^d.

Unigrid point of view. Let us look again at the multi-grid algorithm from the variational standpoint. One natural class of iterative algorithms for minimizing H are the

[18] For a detailed exposition of multi-grid convergence proofs, see [32, Chapters 6–8, 10, 11], [40] and the references cited therein. The additional work needed to handle the piecewise-constant interpolation can be found in [41].

so-called *directional methods*: let $p_0, p_1, \ldots$ be a sequence of "direction vectors" in U, and define $\varphi^{(n+1)}$ to be that vector of the form $\varphi^{(n)} + \lambda p_n$ which minimizes H. The algorithm thus travels "downhill" from $\varphi^{(n)}$ along the line $\varphi^{(n)} + \lambda p_n$ until reaching the minimum of H, then switches to direction p_{n+1} starting from this new point $\varphi^{(n+1)}$, and so on. For a suitable choice of the direction vectors $p_0, p_1, \ldots$, this method can be proven to converge to the global minimum of H [38, pp. 513–520].

Now, some iterative algorithms for minimizing $H(\varphi)$ can be recognized as special cases of the directional method. For example, the Gauss-Seidel iteration is a directional method in which the direction vectors are chosen to be unit vectors $e_1, e_2, \ldots, e_N$ (i.e. vectors which take the value 1 at a single grid point and zero at all others), where $N = \dim U$. [One step of the Gauss-Seidel iteration corresponds to N steps of the directional method.] Similarly, it is not hard to see [39] that the multi-grid iteration with the variational choices of restriction and coarse-grid operators, and with Gauss-Seidel smoothing at each level, is itself a directional method: some of the direction vectors are the unit vectors $e_1^{(M)}, e_2^{(M)}, \ldots, e_{N_M}^{(M)}$ of the fine-grid space, but other direction vectors are the images in the fine-grid space of the unit vectors of the coarse-grid spaces, i.e. they are $p_{M,l}\, e_1^{(l)}, p_{M,l}\, e_2^{(l)}, \ldots, p_{M,l}\, e_{N_l}^{(l)}$. The exact order in which these direction vectors are interleaved depends on the parameters m_1, m_2 and γ which define the cycling structure of the multi-grid algorithm. For example, if $m_1 = 1$, $m_2 = 0$ and $\gamma = 1$, the order of the direction vectors is $\{M\}, \{M-1\}, \ldots, \{0\}$, where $\{l\}$ denotes the sequence $p_{M,l}\, e_1^{(l)}, p_{M,l}\, e_2^{(l)}, \ldots, p_{M,l}\, e_{N_l}^{(l)}$. If $m_1 = 0$, $m_2 = 1$ and $\gamma = 1$, the order is $\{0\}, \{1\}, \ldots, \{M\}$. The reader is invited to work out other cases.

Thus, the multi-grid algorithm (for problems defined in variational form) is a directional method in which the direction vectors include both "single-site modes" $\{M\}$ and also "collective modes" $\{M-1\}, \{M-2\}, \ldots, \{0\}$ on all length scales. For example, if $p_{l,l-1}$ is piecewise-constant injection, then the direction vectors are characteristic functions χ_B (i.e. functions which are 1 on the block $B \subset \Omega$ and zero outside B), where the sets B are successively single sites, cubes of side 2, cubes of side 4, and so on. Similarly, if $p_{l,l-1}$ is linear interpolation, then the direction vectors are triangular waves of various widths.

The multi-grid algorithm has thus an alternative interpretation as a collective-mode algorithm working solely in the fine-grid space U. We emphasize that this "unigrid" viewpoint [39] is mathematically fully equivalent to the recursive definition given earlier. But it gives, we think, an important additional insight into what the multi-grid algorithm is really doing.

For example, for the simple model problem (Poisson equation in a square), we know that the "correct" collective modes are sine waves, in the sense that these modes diagonalize the Laplacian, so that in this basis the Jacobi or Gauss-Seidel algorithm would give the exact solution in a *single* iteration ($M_{Jacobi} = M_{GS} = 0$). On the other hand, the multi-grid method uses square-wave (or triangular-wave) updates, which are not exactly the "correct" collective modes. Nevertheless, the multi-grid convergence proofs [32, 40, 41] assure us that they are "close enough": the norm of the multi-grid iteration matrix M_l is bounded away from 1, uniformly in the lattice size, so that an accurate solution is reached in a *very few* MG iterations (in particular, critical slowing-down is completely eliminated). This viewpoint also explains why MG convergence is more delicate for piecewise-constant interpolation than for piecewise-linear: the point is that a sine wave (or other slowly varying function) can be approximated to arbitrary accuracy (in energy norm) by piecewise-linear functions but *not* by piecewise-constant functions.

We remark that McCormick and Ruge [39] have advocated the "unigrid" idea not just as an alternate point of view on the multi-grid algorithm, but as an alternate *computational procedure*. To be sure, the unigrid method is somewhat simpler to program, and this could have pedagogical advantages. But one of the key properties of the multi-grid method, namely the $O(L^d)$ computational labor per iteration, is sacrificed in the unigrid scheme. Instead of (5.33)–(5.34) one has

$$W_l \approx W_M \tag{5.36}$$

and hence

$$\begin{aligned} work(UG) &\approx W_M \sum_{l=M}^{0} \gamma^{M-l} \\ &\sim \begin{cases} MW_M & \text{if } \gamma = 1 \\ \gamma^M W_M & \text{if } \gamma > 1 \end{cases} \end{aligned} \tag{5.37}$$

Since $M \approx \log_2 L$ and $W_M \sim L^d$, we obtain

$$work(UG) \sim \begin{cases} L^d \log L & \text{if } \gamma = 1 \\ L^{d+\log_2 \gamma} & \text{if } \gamma > 1 \end{cases} \tag{5.38}$$

For a V-cycle the additional factor of $\log L$ is perhaps not terribly harmful, but for a W-cycle the additional factor of L is a severe drawback (though not as severe as the $O(L^2)$ critical slowing-down of the traditional algorithms). Thus, we do *not* advocate the use of unigrid as a computational method if there is a viable multi-grid alternative. Unigrid could, however, be of interest in cases where true multi-grid is unfeasible, as may occur for non-Abelian lattice gauge theories.

Multi-grid algorithms can also be devised for some models in which state space is a nonlinear manifold, such as nonlinear σ-models and lattice gauge theories [20, Sections 3–5]. The simplest case is the XY model: both the fine-grid and coarse-grid field variables are *angles*, and the interpolation operator is piecewise-constant (with angles added modulo 2π). Thus, a coarse-grid variable ψ_y specifies the angle by which the 2^d spins in the block B_y are to be simultaneously rotated. A similar strategy can be employed for nonlinear σ-models taking values in a group G (the so-called "principal chiral models"): the coarse-grid variable ψ_y simultaneously left-multiplies the 2^d spins in the block B_y. For nonlinear σ-models taking values in a nonlinear manifold M on which a group G acts [e.g. the n-vector model with $M = S_{n-1}$ and $G = SO(n)$], the coarse-grid-correction moves are still simultaneous rotation; this means that while the fine-grid fields lie in M, the coarse-grid fields all lie in G. Similar ideas can be applied to lattice gauge theories; the key requirement is to respect the geometric (parallel-transport) properties of the theory. Unfortunately, the resulting algorithms appear to be practical only in the abelian case. (In the non-abelian case, the coarse-grid Hamiltonian becomes too complicated.) Much more work needs to be done on devising good interpolation operators for non-abelian lattice gauge theories.

5.2 Multi-Grid Monte Carlo

Classical equilibrium statistical mechanics is a natural generalization of classical statics (for problems posed in variational form): in the latter we seek to minimize a Hamiltonian $H(\varphi)$, while in the former we seek to generate random samples from the Boltzmann-Gibbs probability distribution $e^{-\beta H(\varphi)}$. The statistical-mechanical problem reduces to the deterministic one in the zero-temperature limit $\beta \to +\infty$.

Likewise, many (but not all) of the deterministic iterative algorithms for minimizing $H(\varphi)$ can be generalized to stochastic iterative algorithms — that is, dynamic Monte Carlo methods — for generating random samples from $e^{-\beta H(\varphi)}$. For example, the stochastic generalization of the Gauss-Seidel algorithm (or more generally, nonlinear Gauss-Seidel with exact minimization) is the single-site heat-bath algorithm; and the stochastic generalization of multi-grid is multi-grid Monte Carlo.

Let us explain these correspondences in more detail. In the Gauss-Seidel algorithm, the grid points are swept in some order, and at each stage the Hamiltonian is minimized as a function of a single variable φ_x, with all other variables $\{\varphi_y\}_{y\neq x}$ being held fixed. The single-site heat-bath algorithm has the same general structure, but the new value φ'_x is chosen randomly from the conditional distribution of $e^{-\beta H(\varphi)}$ given $\{\varphi_y\}_{y\neq x}$, i.e. from the one-dimensional probability distribution

$$P(\varphi'_x)\, d\varphi'_x \;=\; \text{const} \times \exp\left[-\beta H(\varphi'_x, \{\varphi_y\}_{y\neq x})\right]\, d\varphi'_x \tag{5.39}$$

(where the normalizing constant depends on $\{\varphi_y\}_{y\neq x}$). It is not difficult to see that this operation leaves invariant the Gibbs distribution $e^{-\beta H(\varphi)}$. As $\beta \to +\infty$ it reduces to the Gauss-Seidel algorithm.

It is useful to visualize geometrically the action of the Gauss-Seidel and heat-bath algorithms within the space U of all possible field configurations. Starting at the current field configuration φ, the Gauss-Seidel and heat-bath algorithms propose to move the system along the line in U consisting of configurations of the form $\varphi' = \varphi + t\delta_x$ $(-\infty < t < \infty)$, where δ_x denotes the configuration which is 1 at site x and zero elsewhere. In the Gauss-Seidel algorithm, t is chosen so as to minimize the Hamiltonian restricted to the given line; while in the heat-bath algorithm, t is chosen randomly from the the conditional distribution of $e^{-\beta H(\varphi)}$ restricted to the given line, namely the one-dimensional distribution with probability density $P_{cond}(t) \sim \exp[-H_{cond}(t)] \equiv \exp[-H(\varphi + t\delta_x)]$.

The method of *partial resampling* generalizes the heat-bath algorithm in two ways:

1) The "fibers" used by the algorithm need not be lines, but can be higher-dimensional linear or even nonlinear manifolds.

2) The new configuration φ' need not be chosen *independently* of the old configuration φ (as in the heat-bath algorithm); rather, it can be selected by any updating procedure which leaves invariant the conditional probability distribution of $e^{-\beta H(\varphi)}$ restricted to the fiber.

The multi-grid Monte Carlo (MGMC) algorithm is a partial-resampling algorithm in which the "fibers" are the sets of field configurations that can be obtained one from another by a coarse-grid-correction step, i.e. the sets of fields $\varphi + p_{l,l-1}\psi$ with φ fixed and ψ varying over U_{l-1}. These fibers form a family of parallel affine subspaces in U_l, of dimension $N_{l-1} \;=\; \dim U_{l-1}$.

The ingredients of the MGMC algorithm are identical to those of the deterministic MG algorithm, with one exception: the deterministic smoothing iteration $\mathcal{S}_l$ is replaced by a stochastic smoothing iteration (for example, single-site heat-bath). That is, $\mathcal{S}_l(\,\cdot\,, H_l)$ is a stochastic updating procedure $\varphi_l \to \varphi_l'$ that leaves invariant the Gibbs distribution $e^{-\beta H_l}$:

$$\int d\varphi_l \, e^{-\beta H_l(\varphi_l)} \, P_{\mathcal{S}_l(\,\cdot\,,H_l)}(\varphi_l \to \varphi_l') \;=\; d\varphi_l' \, e^{-\beta H_l(\varphi_l')} \,. \tag{5.40}$$

The MGMC algorithm is then defined as follows:

procedure $mgmc(l, \varphi, H_l)$

comment This algorithm updates the field φ in such a way as to leave invariant the probability distribution $e^{-\beta H_l}$.

$\varphi \leftarrow \mathcal{S}_l^{pre}(\varphi, H_l)$

if $l > 0$ **then**

  compute $H_{l-1}(\,\cdot\,) \equiv H_l(\varphi + p_{l,l-1}\,\cdot\,)$

  $\psi \leftarrow 0$

  for $j = 1$ **until** γ_l **do** $mgmc(l-1, \psi, H_{l-1})$

  $\varphi \leftarrow \varphi + p_{l,l-1}\psi$

endif

$\varphi \leftarrow \mathcal{S}_l^{post}(\varphi, H_l)$

end

The alert reader will note that this algorithm is *identical* to the deterministic MG algorithm presented earlier; only the meaning of $\mathcal{S}_l$ is different.

The validity of the MGMC algorithm is proven inductively, starting at level 0 and working upwards. That is, if $mgmc(l-1, \,\cdot\,, H_{l-1})$ is a stochastic updating procedure that leaves invariant the probability distribution $e^{-\beta H_{l-1}}$, then $mgmc(l, \,\cdot\,, H_l)$ leaves invariant $e^{-\beta H_l}$. Note that the coarse-grid-correction step of the MGMC algorithm differs from the heat-bath algorithm in that the new configuration φ' is *not* chosen independently of the old configuration φ; to do so would be impractical, since the fiber has such high dimension. Rather, φ' (or what is equivalent, ψ) is chosen by a *valid updating procedure* — namely, MGMC itself!

The MGMC algorithm has also an alternate interpretation — the *unigrid* viewpoint — in which the fibers are one-dimensional and the resamplings are independent. More precisely, the fibers are lines of the form $\varphi' = \varphi + t\chi_B$ $(-\infty < t < \infty)$, where χ_B denotes the function which is 1 for sites belonging to the block B and zero elsewhere. The sets B are taken successively to be single sites, cubes of side 2, cubes of side 4, and so on. (If linear interpolation were used, then the "direction vectors" χ_B would be replaced by triangular waves of various widths.) Just as the deterministic unigrid algorithm chooses t so as to minimize the "conditional Hamiltonian" $H_{cond}(t) \equiv H(\varphi + t\chi_B)$, so the stochastic unigrid algorithm chooses t randomly from the one-dimensional distribution with probability density $P_{cond}(t) \sim \exp[-H_{cond}(t)]$. Conceptually this algorithm is no more complicated than the single-site heat-bath algorithm. But physically it is of course very different, as the direction vectors χ_B represent *collective modes* on all length scales.

We emphasize that the stochastic unigrid algorithm is *mathematically and physically equivalent* to the multi-grid Monte Carlo algorithm described above. But it is useful, we believe, to be able to look at MGMC from either of the two points of view: independent resamplings in one-dimensional fibers, or non-independent resamplings (defined recursively) in higher-dimensional (coarse-grid) fibers. On the other hand, the two algorithms are not *computationally* equivalent. One MGMC sweep requires a CPU time of order volume (provided that $\gamma < 2^d$), while the time for a unigrid sweep grows faster than the volume [cf. the work estimates (5.34) and (5.38)]. Therefore, we advocate unigrid only as a conceptual device, not as a computational algorithm.

How well does MGMC perform? The answer is highly model-dependent:

- For the *Gaussian model*, it can be proven rigorously [25, 20, 41] that τ is *bounded* as criticality is approached (empirically $\tau \approx 1-2$); therefore, critical slowing-down is *completely eliminated*. The proof is a simple Fock-space argument, combined with the convergence proof for deterministic MG; this will be discussed in Section 5.3.

- For the φ^4 *model*, numerical experiments [25] show that τ diverges with the *same* dynamic critical exponent as in the heat-bath algorithm; the gain in efficiency thus approaches a *constant* factor $F(\lambda)$ near the critical point. This behavior can be understood [20, Section 9.1] as due to the double-well nature of the φ^4 potential, which makes MGMC ineffective on large blocks. Thus, the correct collective modes at long length scales are nonlinear excitations *not* well modelled by $\varphi \to \varphi + t\chi_B$. (See Section 6 for an algorithm that appears to model these excitations well, at least for λ not too small.)

- For the $d = 2$ *XY model*, our numerical data [26] show a more complicated behavior: As the critical temperature is approached from above, τ diverges with a dynamic critical exponent $z = 1.4 \pm 0.3$ for the MGMC algorithm (in either V-cycle or W-cycle), compared to $z = 2.1 \pm 0.3$ for the heat-bath algorithm. Thus, critical slowing-down is significantly reduced but is still very far from being eliminated. On the other hand, below the critical temperature, τ is very small ($\approx 1-2$), uniformly in L and β (at least for the W-cycle); critical slowing-down appears to be completely eliminated. This very different behavior in the two phases can be understood physically: in the low-temperature phase the main excitations are spin waves, which are well handled by MGMC (as in the Gaussian model); but near the critical temperature the important excitations are widely separated vortex-antivortex pairs, which are apparently not easily created by the MGMC updates.

• For the $O(4)$ nonlinear σ-model in two dimensions, which is asymptotically free, preliminary data [42] show a very strong reduction, but not the total elimination, of critical slowing-down. For a W-cycle we find that $z \approx 0.6$ (I emphasize that these data are *very* preliminary!). Previously, Goodman and I had argued heuristically [20, Section 9.3] that for asymptotically free theories with a continuous symmetry group, MGMC (with a W-cycle) should completely eliminate critical slowing-down except for a possible logarithm. But our reasoning may now need to be re-examined![19]

5.3 Stochastic Linear Iterations for the Gaussian Model

In this section we analyze an important class of Markov chains, the stochastic linear iterations for Gaussian models [20, Section 8].[20] This class includes, among others, the single-site heat-bath algorithm (with deterministic sweep of the sites), the stochastic SOR algorithm [44, 45] and the multi-grid Monte Carlo algorithm — all, of course, in the Gaussian case only. We show that the behavior of the stochastic algorithm is completely determined by the behavior of the corresponding deterministic algorithm for solving linear equations. In particular, we show that the exponential autocorrelation time τ_{exp} of the stochastic linear iteration is *equal* to the relaxation time of the corresponding linear iteration.

Consider any quadratic Hamiltonian

$$H(\varphi) \;=\; \frac{1}{2}(\varphi, A\varphi) \;-\; (f, \varphi)\,, \tag{5.41}$$

where A is a symmetric positive-definite matrix. The corresponding Gaussian measure

$$d\pi(\varphi) \;=\; \text{const} \times e^{-\frac{1}{2}(\varphi, A\varphi)+(f,\varphi)}\, d\varphi \tag{5.42}$$

has mean $A^{-1}f$ and covariance matrix A^{-1}. Next consider any first-order stationary linear stochastic iteration of the form

$$\varphi^{(n+1)} \;=\; M\varphi^{(n)} \;+\; Nf \;+\; Q\xi^{(n)}\,, \tag{5.43}$$

[19]**Note Added 1996:** For work on multi-grid Monte Carlo covering the period 1992–96, see [106, 107, 108, 103] and the references cited therein.

[20]Some of this material appears also in [43].

where M, N and Q are fixed matrices and the $\xi^{(n)}$ are independent Gaussian random vectors with mean zero and covariance matrix C. The iteration (5.43) has a unique stationary distribution if and only if the spectral radius $\rho(M) \equiv \lim_{n\to\infty} \|M^n\|^{1/n}$ is < 1; and in this case the stationary distribution is the desired Gaussian measure (5.42) for all f if and only if

$$\begin{aligned} N &= (I-M)A^{-1} && (5.44\text{a}) \\ QCQ^T &= A^{-1} - MA^{-1}M^T && (5.44\text{b}) \end{aligned}$$

(here ^{T} denotes transpose).

The reader will note the close analogy with the deterministic linear problem (5.3)–(5.6). Indeed, (5.44a) is identical with (5.6); and if we take the "zero-temperature limit" in which H is replaced by βH with $\beta \to +\infty$, then the Gaussian measure (5.42) approaches a delta function concentrated at the unique minimum of H (namely, the solution of the linear equation $A\varphi = f$), and the "noise" term disappears ($Q \to 0$), so that the stochastic iteration (5.43) turns into the deterministic iteration (5.5). That is:

(a) The linear deterministic problem is the zero-temperature limit of the Gaussian stochastic problem; and the first-order stationary linear deterministic iteration is the zero-temperature limit of the first-order stationary linear stochastic iteration. Therefore, any stochastic linear iteration for generating samples from the Gaussian measure (5.42) gives rise to a deterministic linear iteration for solving the linear equation (5.3), simply by setting $Q = 0$.

(b) Conversely, the stochastic problem and iteration are the nonzero-temperature generalizations of the deterministic ones. In principle this means that a deterministic linear iteration for solving (5.3) can be generalized to a stochastic linear iteration for generating samples from (5.42), if and only if the matrix $A^{-1} - MA^{-1}M^T$ is positive-semidefinite: just choose a matrix Q satisfying (5.44b). In practice, however, such an algorithm is computationally tractable only if the matrix Q has additional nice properties such as sparsity (or triangularity with a sparse inverse).

Examples. 1. *Single-site heat bath (with deterministic sweep of the sites) = stochastic Gauss-Seidel.* On each visit to site i, φ_i is replaced by a new value φ'_i chosen independently from the conditional distribution of (5.42) with $\{\varphi_j\}_{j\neq i}$ fixed at their current values: that is, φ'_i is Gaussian with mean $(f_i - \sum_{j\neq i} a_{ij}\varphi_j)/a_{ii}$ and variance $1/a_{ii}$. When updating φ_i at sweep $n+1$, the variables φ_j with $j < i$ have already been visited on this sweep, hence have their "new" values $\varphi_j^{(n+1)}$, while the variables φ_j with $j > i$ have not yet been visited on this sweep, and so have their "old" values $\varphi_j^{(n)}$. It follows that

$$\varphi_i^{(n+1)} = a_{ii}^{-1}\left(f_i - \sum_{j<i} a_{ij}\varphi_j^{(n+1)} - \sum_{j>i} a_{ij}\varphi_j^{(n)}\right) + a_{ii}^{-1/2}\xi_i^{(n)}, \qquad (5.45)$$

where ξ has covariance matrix I. A little algebra brings this into the matrix form (5.43) with

$$M = -(D+L)^{-1}L^T \qquad (5.46\text{a})$$

$$
\begin{aligned}
N &= (D+L)^{-1} && (5.46b) \\
Q &= (D+L)^{-1} D^{1/2} && (5.46c)
\end{aligned}
$$

where D and L are the diagonal and lower-triangular parts of the matrix A, respectively. It is straightforward to verify that (5.44a,b) are satisfied.[21] The single-site heat-bath algorithm is clearly the stochastic generalization of the Gauss-Seidel algorithm.

2. *Stochastic SOR.* For models which are Gaussian (or more generally, "multi-Gaussian"), Adler [44] and Whitmer [45] have shown that the successive over-relaxation (SOR) iteration admits a stochastic generalization, namely

$$
\varphi_i^{(n+1)} = (1-\omega)\varphi_i^{(n)} + \omega a_{ii}^{-1} \left(f_i - \sum_{j<i} a_{ij}\varphi_j^{(n+1)} - \sum_{j>i} a_{ij}\varphi_j^{(n)} \right) + \left(\frac{\omega(2-\omega)}{a_{ii}} \right)^{1/2} \xi_i^{(n)} \,, \tag{5.47}
$$

where $0 < \omega < 2$. For $\omega = 1$ this reduces to the single-site heat-bath algorithm. This is easily seen to be of the form (5.43) with

$$
\begin{aligned}
M &= -(D+\omega L)^{-1} \Big[(\omega - 1) D + \omega L^T \Big] && (5.48a) \\
N &= \omega (D+\omega L)^{-1} && (5.48b) \\
Q &= [\omega(2-\omega)]^{1/2} (D+\omega L)^{-1} D^{1/2} && (5.48c)
\end{aligned}
$$

where D and L are as before. It is straightforward to verify that (5.44a,b) are satisfied.[21]

3. *Multi-Grid Monte Carlo (MGMC).* The multi-grid Monte Carlo algorithm *mgmc* (defined in Section 5.2) is identical to the corresponding deterministic multi-grid algorithm *mgm* (defined in Section 5.1) except that $\mathcal{S}_l$ is a stochastic rather than deterministic updating. Consider, for example, the case in which $\mathcal{S}_l$ is a stochastic *linear* updating (e.g. single-site heat-bath). Then the MGMC is also a stochastic linear updating of the form (5.43): in fact, M equals M_{MG}, the iteration matrix of the corresponding deterministic MG method, and N equals N_{MG}; the matrix Q is rather complicated, but since the MGMC algorithm is correct, Q must satisfy (5.44b). [The easiest way to see that $M = M_{MG}$ is to imagine what would happen if all the random numbers $\xi^{(n)}$ were zero. Then the stochastic linear updating would reduce to the corresponding deterministic updating, and hence the same would be true for the MGMC updating as a whole.]

4. *Langevin equation with small time step.* As far as I know, there does not exist any useful stochastic generalization of the Jacobi iteration. However, let us discretize the Langevin equation

$$
\frac{d\varphi}{dt} = -\frac{1}{2} C(A\varphi - f) + \xi \,, \tag{5.49}
$$

[21] We remark that this verification never uses the fact that D is diagonal or that L is lower triangular. It is sufficient to have $A = D + L + L^T$ with D symmetric positive-definite and $D + L$ nonsingular. However, for the method to be practical, it is important that $D^{1/2}$ and $(D+L)^{-1}$ be "easy" to compute when applied to a vector.

where ξ is Gaussian white noise with covariance matrix C, using a small time step δ. The result is an iteration of the form (5.43) with

$$M = I - \frac{\delta}{2} CA \tag{5.50a}$$

$$N = \frac{\delta}{2} C \tag{5.50b}$$

$$Q = \delta^{1/2} I \tag{5.50c}$$

This satisfies (5.44a) exactly, but satisfies (5.44b) only up to an error of order δ. If $C = D^{-1}$, these M, N correspond to a damped Jacobi iteration with $\omega = \delta/2 \ll 1$.

It is straightforward to analyze the dynamic behavior of the stochastic linear iteration (5.43). Using (5.43) and (5.44) to express $\varphi^{(n)}$ in terms of the independent random variables $\varphi^{(0)}, \xi^{(0)}, \xi^{(1)}, \ldots, \xi^{(n-1)}$, we find after a bit of manipulation that

$$\langle \varphi^{(n)} \rangle = M^n \langle \varphi^{(0)} \rangle + (I - M^n) A^{-1} f \tag{5.51}$$

and

$$\operatorname{cov}(\varphi^{(s)}, \varphi^{(t)}) = M^s \operatorname{cov}(\varphi^{(0)}, \varphi^{(0)}) (M^T)^t + \begin{cases} [A^{-1} - M^s A^{-1} (M^T)^s](M^T)^{t-s} & \text{if } s \le t \\ M^{s-t}[A^{-1} - M^t A^{-1} (M^T)^t] & \text{if } s \ge t \end{cases} \tag{5.52}$$

Now let us either start the stochastic process in equilibrium

$$\langle \varphi^{(0)} \rangle = A^{-1} f \tag{5.53a}$$

$$\operatorname{cov}(\varphi^{(0)}, \varphi^{(0)}) = A^{-1} \tag{5.53b}$$

or else let it relax to equilibrium by taking $s, t \to +\infty$ with $s - t$ fixed. Either way, we conclude that in equilibrium (5.43) defines a Gaussian stationary stochastic process with mean $A^{-1} f$ and autocovariance matrix

$$\operatorname{cov}(\varphi^{(s)}, \varphi^{(t)}) = \begin{cases} A^{-1} (M^T)^{t-s} & \text{if } s \le t \\ M^{s-t} A^{-1} & \text{if } s \ge t \end{cases} \tag{5.54}$$

Moreover, since the stochastic process is Gaussian, all higher-order time-dependent correlation functions are determined in terms of the mean and autocovariance. Thus, the matrix M determines the autocorrelation functions of the Monte Carlo algorithm.

Another way to state these relationships is to recall [46, 47] that the Hilbert space $L^2(\pi)$ is isomorphic to the bosonic Fock space $\mathcal{F}(U)$ built on the "energy Hilbert space" (U, A): the "n-particle states" are the homogeneous Wick polynomials of degree n in the shifted field $\tilde{\varphi} = \varphi - A^{-1} f$. (If U is one-dimensional, these are just the Hermite polynomials.) Then the transition probability $P(\varphi^{(n)} \to \varphi^{(n+1)})$ induces on the Fock space an operator

$$P = \Gamma(M^T) \equiv I \oplus M^T \oplus (M^T \otimes M^T) \oplus \cdots \tag{5.55}$$

that is the second quantization of the operator M^T on the energy Hilbert space (see [20, Section 8] for details). It follows from (5.55) that

$$\|\Gamma(M)^n \upharpoonright \mathbf{1}^\perp\|_{L^2(\pi)} \;=\; \|M^n\|_{(U,A)} \tag{5.56}$$

and hence that

$$\rho(\Gamma(M) \upharpoonright \mathbf{1}^\perp) \;=\; \rho(M) \; . \tag{5.57}$$

Moreover, P is self-adjoint on $L^2(\pi)$ [i.e. satisfies detailed balance] if and only if M is self-adjoint with respect to the energy inner product, i.e.

$$MA \;=\; AM^T \; ; \tag{5.58}$$

and in this case

$$\rho(\Gamma(M) \upharpoonright \mathbf{1}^\perp) \;=\; \|\Gamma(M) \upharpoonright \mathbf{1}^\perp\|_{L^2(\pi)} \;=\; \rho(M) \;=\; \|M\|_{(U,A)} \; . \tag{5.59}$$

In summary, we have shown that the dynamic behavior of any stochastic linear iteration is completely determined by the behavior of the corresponding deterministic linear iteration. In particular, the exponential autocorrelation time τ_{exp} (slowest decay rate of any autocorrelation function) is given by

$$\exp(-1/\tau_{exp}) \;=\; \rho(M) \; , \tag{5.60}$$

and this decay rate is achieved by at least one observable which is linear in the field φ. In other words, the (worst-case) convergence rate of the Monte Carlo algorithm is precisely *equal* to the (worst-case) convergence rate of the corresponding deterministic iteration.

In particular, for Gaussian MGMC, the convergence proofs for deterministic multigrid [32, 40, 41] combined with the arguments of the present section prove rigorously that *critical slowing-down is completely eliminated* (at least for a W-cycle). That is, the autocorrelation time τ of the MGMC method is *bounded* as criticality is approached (empirically $\tau \approx 1 - 2$).

6 Swendsen-Wang Algorithms

A very different type of collective-mode algorithm was proposed two years ago[22] by Swendsen and Wang [27] for Potts spin models. Since then, there has been an explosion of work trying to understand why this algorithm works so well (and why it does not work

[22] **Note Added 1996:** Now nearly *ten* years ago! The great interest in algorithms of Swendsen-Wang type (frequently called *cluster algorithms*) has not abated: see the references cited in the "Notes Added" below, plus many others. See also [109, 110] for reviews that are slightly more up-to-date than the present notes (1990 rather than 1989).

even better), and trying to improve or generalize it. The basic idea behind all algorithms of Swendsen-Wang type is to augment the given model by means of *auxiliary variables*, and then to simulate this augmented model. In this lecture we describe the Swendsen-Wang (SW) algorithm and review some of the proposed variants and generalizations.

Let us first recall that the q-state Potts model [48, 49] is a generalization of the Ising model in which each spin σ_i can take q distinct values rather than just two ($\sigma_i = 1, 2, \ldots, q$); here q is an integer ≥ 2. Neighboring spins prefer to be in the same state, and pay an energy price if they are not. The Hamiltonian is therefore

$$H(\sigma) \;=\; \sum_{\langle ij \rangle} J_{ij}\,(1 - \delta_{\sigma_i,\sigma_j}) \tag{6.1}$$

with $J_{ij} \geq 0$ for all i, j ("ferromagnetism"), and the partition function is

$$\begin{aligned} Z &= \sum_{\{\sigma\}} \exp[-H(\sigma)] \\ &= \sum_{\{\sigma\}} \exp\left[\sum_{\langle ij \rangle} J_{ij}\,(\delta_{\sigma_i,\sigma_j} - 1)\right] \\ &= \sum_{\{\sigma\}} \prod_{\langle ij \rangle} [(1 - p_{ij}) \;+\; p_{ij}\delta_{\sigma_i,\sigma_j}] \end{aligned} \tag{6.2}$$

where we have defined $p_{ij} = 1 - \exp(-J_{ij})$. The Gibbs measure $\mu_{Potts}(\sigma)$ is, of course,

$$\begin{aligned} \mu_{Potts}(\sigma) &= Z^{-1} \exp\left[\sum_{\langle ij \rangle} J_{ij}\,(\delta_{\sigma_i,\sigma_j} - 1)\right] \\ &= Z^{-1} \prod_{\langle ij \rangle} [(1 - p_{ij}) \;+\; p_{ij}\delta_{\sigma_i,\sigma_j}] \end{aligned} \tag{6.3}$$

We now use the deep identity

$$a + b \;=\; \sum_{n=0}^{1} [a\delta_{n,0} \;+\; b\delta_{n,1}] \tag{6.4}$$

on each bond $\langle ij \rangle$; that is, we introduce on each bond $\langle ij \rangle$ an auxiliary variable n_{ij} taking the values 0 and 1, and obtain

$$Z \;=\; \sum_{\{\sigma\}} \sum_{\{n\}} \prod_{\langle ij \rangle} [(1 - p_{ij})\,\delta_{n_{ij},0} \;+\; p_{ij}\delta_{n_{ij},1}\delta_{\sigma_i,\sigma_j}]\,. \tag{6.5}$$

Let us now take seriously the $\{n\}$ as dynamical variables: we can think of n_{ij} as an *occupation variable* for the bond $\langle ij \rangle$ (1 = occupied, 0 = empty). We therefore define the *Fortuin-Kasteleyn-Swendsen-Wang (FKSW) model* to be a joint model having q-state Potts spins σ_i at the sites and occupation variables n_{ij} on the bonds, with joint probability distribution

$$\mu_{FKSW}(\sigma, n) \;=\; Z^{-1} \prod_{\langle ij \rangle} [(1 - p_{ij})\,\delta_{n_{ij},0} \;+\; p_{ij}\delta_{n_{ij},1}\delta_{\sigma_i,\sigma_j}]\,. \tag{6.6}$$

Finally, let us see what happens if we sum over the $\{\sigma\}$ at fixed $\{n\}$. Each occupied bond $\langle ij \rangle$ imposes a constraint that the spins σ_i and σ_j must be in the same state, but otherwise the spins are unconstrained. We therefore group the sites into connected clusters (two sites are in the same cluster if they can be joined by a path of occupied bonds); then all the spins within a cluster must be in the same state (all q values are equally probable), and distinct clusters are independent. It follows that

$$Z \;=\; \sum_{\{n\}} \left(\prod_{\langle ij \rangle :\, n_{ij}=1} p_{ij} \right) \left(\prod_{\langle ij \rangle :\, n_{ij}=0} (1-p_{ij}) \right) q^{\mathcal{C}(n)} \;, \tag{6.7}$$

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

$$\mu_{RC}(n) \;=\; Z^{-1} \left(\prod_{\langle ij \rangle :\, n_{ij}=1} p_{ij} \right) \left(\prod_{\langle ij \rangle :\, n_{ij}=0} (1-p_{ij}) \right) q^{\mathcal{C}(n)} \;, \tag{6.8}$$

is called the *random-cluster model with parameter* q. This is a generalized bond-percolation model, with non-local correlations coming from the factor $q^{\mathcal{C}(n)}$; for $q = 1$ it reduces to ordinary bond percolation. Note, by the way, that in the random-cluster model (unlike the Potts and FKSW models), q is merely a parameter; it can take any positive real value, not just $2, 3, \ldots$. So the random-cluster model defines, in some sense, an analytic continuation of the Potts model to non-integer q; ordinary bond percolation corresponds to the "one-state Potts model".

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

a) $Z_{Potts} = Z_{FKSW} = Z_{RC}$.

b) The marginal distribution of μ_{FKSW} on the Potts variables $\{\sigma\}$ (integrating out the $\{n\}$) is precisely the Potts model $\mu_{Potts}(\sigma)$.

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

The conditional distributions of μ_{FKSW} are also simple:

d) The conditional distribution of the $\{n\}$ given the $\{\sigma\}$ is as follows: independently for each bond $\langle ij \rangle$, one sets $n_{ij} = 0$ in case $\sigma_i \neq \sigma_j$, and sets $n_{ij} = 0, 1$ with probability $1 - p_{ij}, p_{ij}$, respectively, in case $\sigma_i = \sigma_j$.

e) The conditional distribution of the $\{\sigma\}$ given the $\{n\}$ is as follows: independently for each connected cluster, one sets all the spins σ_i in the cluster to the same value, chosen equiprobably from $\{1, 2, \ldots, q\}$.

These facts can be used for both analytic and numerical purposes. For example, by using facts (b), (c) and (e) we can prove an identity relating expectations in the Potts model to connection probabilities in the random-cluster model:

$$
\begin{aligned}
\langle \delta_{\sigma_i,\sigma_j} \rangle_{Potts,q} &= \langle \delta_{\sigma_i,\sigma_j} \rangle_{FKSW,q} && \text{[by (b)]} \\
&= \langle E(\delta_{\sigma_i,\sigma_j} \,|\, \{n\}) \rangle_{FKSW,q} \\
&= \left\langle \frac{(q-1)\gamma_{ij}+1}{q} \right\rangle_{FKSW,q} && \text{[by (e)]} \\
&= \left\langle \frac{(q-1)\gamma_{ij}+1}{q} \right\rangle_{RC,q} && \text{[by (c)]}
\end{aligned}
\tag{6.9}
$$

Here

$$
\gamma_{ij} \;\equiv\; \gamma_{ij}(n) \;\equiv\; \begin{cases} 1 & \text{if } i \text{ is connected to } j \\ 0 & \text{if } i \text{ is not connected to } j \end{cases}
\tag{6.10}
$$

and $E(\,\cdot\,|\,\{n\})$ denotes conditional expectation given $\{n\}$.[23] For the Ising model with the usual convention $\sigma = \pm 1$, (6.9) can be written more simply as

$$
\langle \sigma_i \sigma_j \rangle_{Ising} \;=\; \langle \gamma_{ij} \rangle_{RC,q=2} \;.
\tag{6.11}
$$

Similar identities can be proven for higher-order correlation functions, and can be employed to prove Griffiths-type correlation inequalities for the Potts model [51, 52].

On the other hand, Swendsen and Wang [27] exploited facts (b)–(e) to devise a radically new type of Monte Carlo algorithm. The Swendsen-Wang algorithm (SW) simulates the joint model (6.6) by alternately applying the conditional distributions (d) and (e) — that is, by alternately generating new bond occupation variables (independent of the old ones) given the spins, and new spin variables (independent of the old ones) given the bonds. Each of these operations can be carried out in a computer time of order volume: for generating the bond variables this is trivial, and for generating the spin variable it relies on efficient (linear-time) algorithms for computing the connected clusters.[24] It is trivial that the SW algorithm leaves invariant the Gibbs measure (6.6), since any product of conditional probability operators has this property. It is also easy to see that the algorithm is ergodic, in the sense that every configuration $\{\sigma, n\}$ having

[23] For an excellent introduction to conditional expectations, see [50].

[24] Determining the connected components of an undirected graph is a classic problem of computer science. The depth-first-search and breadth-first-search algorithms [53] have a running time of order V, while the Fischer-Galler-Hoshen-Kopelman algorithm (in one of its variants) [54] has a worst-case running time of order $V \log V$, and an observed mean running time of order V in percolation-type problems [55]. Both of these algorithms are non-vectorizable. Shiloach and Vishkin [56] have invented a SIMD parallel algorithm, and we have very recently vectorized it for the Cyber 205, obtaining a speedup of a factor of 11 over scalar mode. We are currently carrying out a comparative test of these three algorithms, as a function of lattice size and bond density [57]. In view of the extraordinary performance of the SW algorithm (see below) and the fact that virtually all its CPU time is spent finding connected components, we feel that the desirability of finding improved algorithms for this problem is self-evident.

$d = 2$ Ising Model		
L	χ	$\tau_{int,\mathcal{E}}$
64	1575 (10)	5.25 (0.30)
128	5352 (53)	7.05 (0.67)
256	17921 (109)	6.83 (0.40)
512	59504 (632)	7.99 (0.81)

Table 1: Susceptibility χ and autocorrelation time $\tau_{int,\mathcal{E}}$ ($\mathcal{E}$ = energy $\approx$ slowest mode) for two-dimensional Ising model at criticality, using Swendsen-Wang algorithm. Standard error is shown in parentheses.

nonzero μ_{FKSW}-measure is accessible from every other. So the SW algorithm is at least a *correct* algorithm for simulating the FKSW model. It is also an algorithm for simulating the Potts and random-cluster models, since expectations in these two models are equal to the corresponding expectations in the FKSW model.

Historical remark. The random-cluster model was introduced in 1969 by Fortuin and Kasteleyn [58]; they derived the identity $Z_{Potts} = Z_{RC}$, along with the correlation-function identity (6.9) and some generalizations. These relations were rediscovered several times during the subsequent two decades [59]. Surprisingly, however, no one seems to have noticed the *joint* probability distribution μ_{FKSW} that underlay all these identities; this was discovered implicitly by Swendsen and Wang [27], and was made explicit by Edwards and Sokal [60].

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

How well does the SW algorithm perform? In at least some cases, the performance is nothing short of extraordinary. Table 1 shows some preliminary data [61] on a two-dimensional Ising model at the bulk critical temperature. These data are consistent with the estimate $\tau_{SW} \sim L^{\approx 0.35}$ [27].[25] By contrast, the conventional single-spin-flip

[25] But *precisely because* τ rises so slowly with L, good estimates of the dynamic critical exponent will require the use of *extremely* large lattices. Even with lattices up to $L = 512$, we are unable to distinguish convincingly between $z \approx 0.35$ and $z \approx 0$. **Note Added 1996:** For more recent and

	Estimates of z_{SW}			
	$q=1$	$q=2$	$q=3$	$q=4$
$d=1$	0	0	0	0
$d=2$	0	≈ 0.35	0.55 ± 0.03	≈ 1 (exact??)
$d=3$	0	≈ 0.75	—	—
$d=4$	0	1 (exact?)	—	—

Table 2: Current best estimates of the dynamic critical exponent z for the Swendsen-Wang algorithm. Estimates are taken from [27] for $d = 2, 3$, $q = 2$; [62] for $d = 2$, $q = 3, 4$; and [63] for $d = 4$, $q = 2$. Error bar is a 95% confidence interval.

algorithms for the two-dimensional Ising model have $\tau_{conv} \sim L^{\approx 2.1}$ [21]. So the advantage of Swendsen-Wang over conventional algorithms (for this model) grows asymptotically like $L^{\approx 1.75}$. To be sure, one iteration of the Swendsen-Wang algorithm may be a factor of ~ 10 more costly in CPU time than one iteration of a conventional algorithm (the exact factor depends on the efficiency of the cluster-finding subroutine). But the SW algorithm wins already for modest values of L.

For other Potts models, the performance of the SW algorithm is less spectacular than for the two-dimensional Ising model, but it is still very impressive. In Table 2 we give the current best estimates of the dynamic critical exponent z_{SW} for q-state Potts models in d dimensions, as a function of q and d.[26]

All these exponents are much lower than the $z \gtrsim 2$ observed in the single-spin-flip algorithms.

Although the SW algorithm performs impressively well, we understand very little about *why* these exponents take the values they do. Some cases are easy. If $q = 1$, then all spins are in the same state (the *only* state!), and all bonds are thrown independently, so the autocorrelation time is zero. (Here the SW algorithm just reduces to the standard *static* algorithm for independent bond percolation.) If $d = 1$ (more generally, if the lattice is a *tree*), the SW dynamics is exactly soluble: the behavior of each bond is independent of each other bond, and $\tau_{exp} \to -1/\log(1 - 1/q) < \infty$ as $\beta \to +\infty$. But the remainder of our understanding is very murky. Two principal insights have been obtained so far:

a) A calculation yielding $z_{SW} = 1$ in a mean-field (Curie-Weiss) Ising model [64].

much more precise data, see [111, 112]. These data are consistent with $\tau_{SW} \sim L^{\approx 0.24}$, but they are also consistent with $\tau_{SW} \sim \log^2 L$ [112]. It is *extremely* difficult to distinguish a small power from a logarithm.

[26]**Note Added 1996:** For more recent data, see [111, 112] for $d = 2$, $q = 2$; [113] for $d = 2$, $q = 3$; and [112, 114] for $d = 2$, $q = 4$. The situation for $d = 3$, $q = 2$ is extremely unclear: exponent estimates by different workers (using different methods) disagree wildly.

This suggests (but of course does not prove) that $z_{SW} = 1$ for Ising models ($q = 2$) in dimension $d \geq 4$.

b) A rigorous proof that $z_{SW} \geq \alpha/\nu$ [62]. This bound, while valid for all d and q, is extremely far from sharp for the Ising models in dimensions 3 and higher. But it is reasonably good for the 3- and 4-state Potts models in two dimensions, and in the latter case it may even be sharp.[27]

But much remains to be understood!

The Potts model with q *large* behaves very differently. Instead of a critical point, the model undergoes a *first-order* phase transition: in two dimensions, this occurs when $q > 4$, while in three or more dimensions, it is believed to occur already when $q \geq 3$ [49]. At a first-order transition, *both* the conventional algorithms and the Swendsen-Wang algorithm have an extremely severe slowing-down (*much* more severe than the slowing-down at a critical point): right at the transition temperature, we expect $\tau \sim \exp(cL^{d-1})$. This is because sets of configurations typical of the ordered and disordered phases are separated by free-energy barriers of order L^{d-1}, i.e. by sets of intermediate configurations that contain interfaces of surface area $\sim L^{d-1}$ and therefore have an equilibrium probability $\sim \exp(-cL^{d-1})$.[28]

Wolff [65] has proposed a interesting modification of the SW algorithm, in which one builds only a *single* cluster (starting at a randomly chosen site) and flips it. Clearly, one step of the single-cluster SW algorithm makes less change in the system than one step of the standard SW algorithm, but it also takes much less work. If one enumerates the cluster using depth-first-search or breadth-first-search, then the CPU time is proportional to the size of the cluster; and by the Fortuin-Kasteleyn identity (6.9), the mean cluster size is equal to the susceptibility:

$$\sum_j \langle \gamma_{ij} \rangle \;=\; \sum_j \left\langle \frac{q\delta_{\sigma_i,\sigma_j} - 1}{q-1} \right\rangle \;\equiv\; \chi \,. \tag{6.12}$$

So the relevant quantity in the single-cluster SW algorithm is the dynamic critical exponent measured *in CPU units*:

$$z_{1-cluster,CPU} \;=\; z_{1-cluster} - \left(d - \frac{\gamma}{\nu} \right) . \tag{6.13}$$

---

[27] **Note Added 1996:** See [112, 113, 114] for more recent data on the possible sharpness of the Li–Sokal bound for two-dimensional models with $q = 2, 3, 4$. These data appear to be consistent with sharpness *modulo a logarithm*, i.e. $\tau_{SW}/C_H \sim \log L$.

[28] **Note Added 1996:** Great progress has been made in recent years in Monte Carlo methods for systems undergoing a first-order phase transition. The key idea is to simulate a suitably chosen *non*-Boltzmann probability distribution ("umbrella", "multicanonical", ...) and then apply reweighting methods. The exponential slowing-down $\tau \sim \exp(cL^{d-1})$ resulting from barrier penetration is replaced by a power-law slowing-down $\tau \sim L^p$ resulting from diffusion, *provided that* the distribution to be simulated is suitably chosen. See [115, 116, 117, 118, 119] for reviews.

The value of the single-cluster algorithm is that the probability of choosing a cluster is proportional to its size (since we pick a random *site*), so the work is concentrated preferentially on larger clusters — and we think that is these clusters which are most important near the critical point. So it would not be surprising if $z_{1-cluster,CPU}$ were smaller than z_{SW}. Preliminary measurements indicate that $z_{1-cluster,CPU}$ is about the same as z_{SW} for the two-dimensional Ising model, but is significantly smaller for the three- and four-dimensional Ising models [66]. But a convincing theoretical understanding of this behavior is lacking.

Several other generalizations of the SW algorithm for Potts models have been proposed.[29] One is a multi-grid extension of the SW algorithm: the idea is to carry out only a partial FKSW transformation, but then to apply this concept recursively [67]. This algorithm *may* have a dynamic critical exponent that is smaller than that of standard SW (but the claims that $z = 0$ are in my opinion unconvincing).[30] A second generalization, which works only in *two* dimensions, augments the SW algorithm by transformations to the dual lattice [68]. This algorithm appears to achieve a modest improvement in critical slowing-down in the scaling region $|\beta - \beta_c| \sim L^{-1/\nu}$.

Finally, the SW algorithm can be generalized in a straightforward manner to Potts lattice gauge theories (more precisely, lattice gauge theories with a *finite abelian* gauge group G and *Potts (δ-function) action*). Preliminary results for the three-dimensional Z_2 gauge theory yield a dynamic critical exponent roughly equal to that of the ordinary SW algorithm for the three-dimensional Ising model (to which the gauge theory is dual) [69].

In the past year there has been a flurry of papers trying to generalize the SW algorithm to non-Potts models. Interesting proposals for a direct generalization of the SW algorithm were made by Edwards and Sokal [60] and by Niedermayer [70]. But the most promising ideas at present seem to be the *embedding algorithms* proposed by Brower and Tamayo [71] for one-component φ^4 models and by Wolff [65, 72] for multi-component $O(n)$-invariant models.

The idea of the embedding algorithms is to find Ising-like variables underlying a general spin variable, and then to update the resulting Ising model using the ordinary SW algorithm (or the single-cluster variant). For one-component spins, this embedding is the obvious decomposition into magnitude and sign. Consider, therefore, a one-

[29]**Note Added 1996:** In addition to the generalizations discussed below, see [120, 112] for SW-type algorithms for the Ashkin-Teller model; see [121] for a clever SW-type algorithm for the fully frustrated Ising model; and see [122, 123, 124, 100] for a very interesting SW-type algorithm for *antiferromagnetic* Potts models, based on the "embedding" idea to be described below.

[30]**Note Added 1996:** For at least some versions of the multi-grid SW algorithm, it can be proven [125] that the bound $z_{MGSW} \geq \alpha/\nu$ holds. Thus, the critical slowing-down is *not* completely eliminated in any model in which the specific heat is divergent.

component model with Hamiltonian

$$H(\varphi) \;=\; -\beta \sum_{\langle xy \rangle} \varphi_x \cdot \varphi_y \;+\; \sum_x V(\varphi_x) \;, \tag{6.14}$$

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

$$\varphi_x \;=\; \varepsilon_x \, |\varphi_x| \;, \tag{6.15}$$

where the signs $\{\varepsilon\}$ are Ising variables. For *fixed* values of the magnitudes $\{|\varphi|\}$, the conditional probability distribution of the $\{\varepsilon\}$ is given by an Ising model with *ferromagnetic* (though space-dependent) couplings $J_{xy} \equiv \beta |\varphi_x| |\varphi_y|$. Therefore, the $\{\varepsilon\}$ model can be updated by the Swendsen-Wang algorithm. (Heat-bath or MGMC sweeps must also be performed, in order to update the magnitudes.) For the two-dimensional φ^4 model, Brower and Tamayo [71] find a dynamic critical behavior identical to that of the SW algorithm for the two-dimensional Ising model — just as one would expect based on the idea that the "important" collective modes in the φ^4 model are spin flips.

Wolff's embedding algorithm for n-component models ($n \geq 2$) is equally simple. Consider an $O(n)$-invariant model with Hamiltonian

$$H(\boldsymbol{\sigma}) \;=\; -\beta \sum_{\langle xy \rangle} \boldsymbol{\sigma}_x \cdot \boldsymbol{\sigma}_y \;+\; \sum_x V(|\boldsymbol{\sigma}_x|) \;, \tag{6.16}$$

with $\beta \geq 0$. Now fix a unit vector $\mathbf{r} \in \mathbb{R}^n$; then any spin vector $\boldsymbol{\sigma}_x \in \mathbb{R}^n$ can be written uniquely (except for a set of measure zero) in the form

$$\boldsymbol{\sigma}_x \;=\; \boldsymbol{\sigma}_x^{\perp} + \varepsilon_x |\boldsymbol{\sigma}_x \cdot \mathbf{r}| \mathbf{r} \;, \tag{6.17}$$

where

$$\boldsymbol{\sigma}_x^{\perp} \;=\; \boldsymbol{\sigma}_x - (\boldsymbol{\sigma}_x \cdot \mathbf{r})\mathbf{r} \tag{6.18}$$

is the component of $\boldsymbol{\sigma}_x$ perpendicular to $\mathbf{r}$, and

$$\varepsilon_x \;=\; \operatorname{sgn}(\boldsymbol{\sigma}_x \cdot \mathbf{r}) \tag{6.19}$$

takes the values ± 1. Therefore, for *fixed* values of the $\{\boldsymbol{\sigma}^{\perp}\}$ and $\{|\boldsymbol{\sigma} \cdot \mathbf{r}|\}$, the probability distribution of the $\{\varepsilon\}$ is given by an Ising model with *ferromagnetic* couplings $J_{xy} \equiv \beta |\boldsymbol{\sigma}_x \cdot \mathbf{r}| |\boldsymbol{\sigma}_y \cdot \mathbf{r}|$. The algorithm is then: Choose at random a unit vector $\mathbf{r}$; fix the $\{\boldsymbol{\sigma}^{\perp}\}$ and $\{|\boldsymbol{\sigma} \cdot \mathbf{r}|\}$ at their current values, and update the $\{\varepsilon\}$ by the standard Swendsen-Wang algorithm (or the single-cluster variant). Flipping ε_x corresponds to reflecting $\boldsymbol{\sigma}_x$ in the hyperplane perpendicular to $\mathbf{r}$.

At first thought it may seem strange (and somehow "unphysical") to try to find Ising-like (i.e. discrete) variables in a model with a *continuous* symmetry group. However, upon reflection (pardon the pun) one sees what is going on: if the spin configuration is slowly varying (e.g. a long-wavelength spin wave), then the clusters tend to break along the surfaces where J_{xy} is small, hence where $\boldsymbol{\sigma} \cdot \mathbf{r} \approx 0$. Then flipping ε_x on some clusters

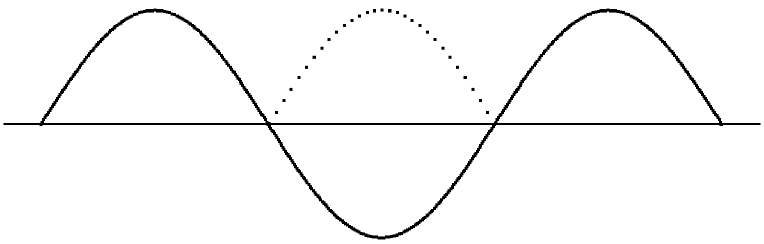

Figure 4: Action of the Wolff embedding algorithm on a long-wavelength spin wave. For simplicity, both spin space ($\boldsymbol{\sigma}$) and physical space (x) are depicted as one-dimensional.

corresponds to a "soft" change near the cluster boundaries but a "hard" change in the cluster interiors, i.e. a long-wavelength collective mode (Figure 4). So it is conceivable that the algorithm could have very small (or even zero!) critical slowing-down in models where the important large-scale collective modes are spin waves.

Even more strikingly, consider a two-dimensional XY-model configuration consisting of a widely separated vortex-antivortex pair: in the continuum limit this is given by

$$\theta(z) = \operatorname{Im}\log(z-a) - \operatorname{Im}\log(z+a) \tag{6.20}$$

where $z \equiv x_1 + ix_2$ and $\boldsymbol{\sigma} \equiv (\cos\theta, \sin\theta)$. Then the surface $\boldsymbol{\sigma}\cdot\mathbf{r} = 0$ is an ellipse passing *directly through* the vortex and antivortex[31]. Reflecting the spins inside this ellipse produces a configuration $\{\boldsymbol{\sigma}_{new}\}$ that is a continuous map of the doubly punctured plane into the semicircle $|\boldsymbol{\sigma}| = 1$, $\boldsymbol{\sigma}\cdot\mathbf{r} \geq 0$. Such a map necessarily has zero winding number around the points $\pm a$. So the Wolff update has destroyed the vorticity![32]

Therefore, the key collective modes in the two-dimensional XY and $O(n)$ models — spin waves and (for the XY case) vortices — are well "encoded" by the Ising variables $\{\varepsilon\}$. So it is quite plausible that critical slowing-down could be eliminated or almost eliminated. In fact, tests of this algorithm on the two-dimensional XY ($n = 2$), classical Heisenberg ($n = 3$) and $O(4)$ models are consistent with the complete absence of critical slowing-down [72, 73].

In view of the extraordinary success of the Wolff algorithm for spin models, it is tempting to try to extend it to lattice gauge theories with continuous gauge group [for example, $U(1)$, $SU(N)$ or $SO(N)$]. But I have nothing to report at present![33]

[31] With $\mathbf{r} = (\cos\phi, \sin\phi)$, the equation of this ellipse is $x_1^2 + (x_2 + a\tan\phi)^2 = a^2\sec^2\phi$.

[32] This picture of the action of the Wolff algorithm on vortex-antivortex pairs was developed in discussions with Richard Brower and Robert Swendsen.

[33] **Note Added 1996:** See [126] for a general theory of Wolff-type embedding algorithms as applied to nonlinear σ-models (spin models taking values in a Riemannian manifold M). Roughly speaking, we argue that a Wolff-type embedding algorithm can work well — in the sense of having a dynamic

7 Algorithms for the Self-Avoiding Walk[34]

An *N–step self-avoiding walk* ω on a lattice $\mathcal{L}$ is a sequence of *distinct* points $\omega_0, \omega_1, \ldots, \omega_N$ in $\mathcal{L}$ such that each point is a nearest neighbor of its predecessor. Let c_N [resp. $c_N(x)$] be the number of N-step SAWs starting at the origin and ending anywhere [resp. ending at x]. Let $\langle \omega_N^2 \rangle$ be the mean-square end-to-end distance of an N-step SAW. These quantities are believed to have the asymptotic behavior

$$
\begin{aligned}
c_N &\sim \mu^N N^{\gamma-1} && (7.1)\\
c_N(x) &\sim \mu^N N^{\alpha_{\rm sing}-2} \qquad (x \text{ fixed } \neq 0) && (7.2)\\
\langle \omega_N^2 \rangle &\sim N^{2\nu} && (7.3)
\end{aligned}
$$

as $N \to \infty$; here γ, $\alpha_{\rm sing}$ and ν are critical exponents, while μ (the connective constant of the lattice) is the analogue of a critical temperature. The SAW has direct application in polymer physics [74], and is indirectly relevant to ferromagnetism and quantum field theory by virtue of its equivalence with the $n \to 0$ limit of the n-vector model [75].

The SAW has some advantages over spin systems for Monte Carlo work: Firstly, one can work directly with SAWs on an infinite lattice; there are no systematic errors due to finite-volume corrections. Secondly, there is no L^d (or ξ^d) factor in the computational work, so one can go closer to criticality. Thus, the SAW is an exceptionally advantageous "laboratory" for the numerical study of critical phenomena.

Different aspects of the SAW can be probed in three different ensembles[35]:

- Free-endpoint grand canonical ensemble (variable N, variable x)
- Fixed-endpoint grand canonical ensemble (variable N, fixed x)
- Canonical ensemble (fixed N, variable x)

In the remainder of this section we survey some typical Monte Carlo algorithms for these ensembles.

Free-endpoint grand canonical ensemble. Here the configuration space S is the set of all SAWs, of arbitrary length, starting at the origin and ending anywhere. The grand

critical exponent z significantly less than 2 — *only if* M is a discrete quotient of a product of spheres: for example, a sphere S^{N-1} or a real projective space $RP^{N-1} \equiv S^{N-1}/Z_2$. In particular, we do *not* expect Wolff-type embedding algorithms to work well for σ-models taking values in $SU(N)$ for $N \geq 3$.

[34]**Note Added 1996:** An extensive and up-to-date review of Monte Carlo algorithms for the self-avoiding walk can be found in [127]. A briefer version is [128].

[35]The proper terminology for these ensembles is unclear to me. Perhaps the grand canonical and canonical ensembles ought to be called "canonical" and "microcanonical", respectively, reserving the term "grand canonical" for ensembles of *many* SAWs. **Note Added 1996:** I now prefer calling these the "variable-length" and "fixed-length" ensembles, respectively, and avoiding the "–canonical" terms entirely; this avoids ambiguity.

partition function is

$$\Xi(\beta) = \sum_{N=0}^{\infty} \beta^N c_N \tag{7.4}$$

and the Gibbs measure is

$$\pi(\omega) = \Xi(\beta)^{-1} \times \beta^{|\omega|}. \tag{7.5}$$

The "monomer activity" β is a user-chosen parameter satisfying $0 \leq \beta < \beta_c \equiv \mu^{-1}$. As β approaches the critical activity β_c, the average walk length $\langle N \rangle$ tends to infinity. The connective constant μ and the critical exponent γ can be estimated from the Monte Carlo data, using the method of maximum likelihood [76].

A dynamic Monte Carlo algorithm for this ensemble was proposed by Berretti and Sokal [76]. The algorithm's elementary moves are as follows: either one attempts to append a new step to the walk, with equal probability in each of the q possible directions (here q is the coordination number of the lattice); or else one deletes the last step from the walk. In the former case, one must check that the proposed new walk is self-avoiding; if it isn't, then the attempted move is rejected and the old configuration is counted again in the sample ("null transition"). If an attempt is made to delete a step from an already-empty walk, then a null transition is also made. The relative probabilities of $\Delta N = +1$ and $\Delta N = -1$ attempts are chosen to be

$$\mathbb{P}(\Delta N = +1 \text{ attempt}) \;=\; \frac{q\beta}{1+q\beta} \tag{7.6}$$

$$\mathbb{P}(\Delta N = -1 \text{ attempt}) \;=\; \frac{1}{1+q\beta} \tag{7.7}$$

Therefore, the transition probability matrix is

$$p(\omega \to \omega') \;=\; \begin{cases} \frac{\beta}{1+q\beta}\chi_{\mathrm{SAW}}(\omega') & \text{if } \omega \prec \omega' \\ \frac{1}{1+q\beta} & \text{if } \omega' \prec \omega \text{ or } \omega = \omega' = \emptyset \\ \frac{\beta}{1+q\beta} A(\omega) & \text{if } \omega = \omega' = \emptyset \end{cases} \tag{7.8}$$

where

$$\chi_{\mathrm{SAW}}(\omega') \;=\; \begin{cases} 1 & \text{if } \omega' \text{ is a SAW} \\ 0 & \text{if } \omega' \text{ is not a SAW} \end{cases} \tag{7.9}$$

Here $\omega \prec \omega'$ denotes that the walk ω' is a one-step extension of ω; and $A(\omega)$ is the number of *non-self-avoiding* walks ω' with $\omega \prec \omega'$. It is easily verified that this transition matrix satisfies detailed balance for the Gibbs distribution π, i.e.

$$\pi(\omega)\, p(\omega \to \omega') \;=\; \pi(\omega')\, p(\omega' \to \omega) \tag{7.10}$$

for all $\omega, \omega' \in S$. It is also easily verified that the algorithm is ergodic (= irreducible): to get from a walk ω to another walk ω', it suffices to use $\Delta N = -1$ moves to transform ω into the empty walk $\emptyset$, and then use $\Delta N = +1$ moves to build up the walk ω'.

Let us now analyze the critical slowing-down of the Berretti-Sokal algorithm. We can argue heuristically that

$$\tau \;\sim\; \langle N \rangle^2 \;. \tag{7.11}$$

To see this, consider the quantity $N(t) = |\omega|(t)$, the number of steps in the walk at time t. This quantity executes, crudely speaking, a random walk (with drift) on the nonnegative integers; the average time to go from some point N to the point 0 (i.e. the empty walk) is of order N^2. Now, each time the empty walk is reached, all memory of the past is erased; future walks are then independent of past ones. Thus, the autocorrelation time ought to be of order $\langle N^2 \rangle$, or equivalently $\langle N \rangle^2$.

This heuristic argument can be turned into a rigorous proof of a *lower bound* $\tau \geq \text{const} \times \langle N \rangle^2$ [10]. However, as an argument for an *upper bound* of the same form, it is not entirely convincing, as it assumes without proof that the *slowest* mode is the one represented by $N(t)$. With considerably more work, it is possible to prove an upper bound on τ that is only slightly weaker than the heuristic prediction: $\tau \leq \text{const} \times \langle N \rangle^{1+\gamma}$ [10, 77].[36] (Note that the critical exponent γ is believed to equal 43/32 in dimension $d = 2$, ≈ 1.16 in $d = 3$, and 1 in $d \geq 4$.) In fact, we suspect [78] that the true behavior is $\tau \sim \langle N \rangle^p$ for some exponent p *strictly between* 2 and $1 + \gamma$. A deeper understanding of the dynamic critical behavior of the Berretti-Sokal algorithm would be of definite value.

It is worth comparing the computational work required for SAW versus Ising simulations: $\langle N \rangle^{\approx 2} \sim \xi^{\approx 2/\nu} = \xi^{\approx 3.4}$ for the $d = 3$ SAW, versus $\xi^{d+z} = \xi^{\approx 5.0}$ (resp.$\xi^{\approx 3.8}$) for the $d = 3$ Ising model using the Metropolis (resp. Swendsen-Wang) algorithm. This vindicates our assertion that the SAW is an advantageous model for Monte Carlo studies of critical phenomena.

Fixed-endpoint grand canonical ensemble. The configuration space S is the set of all SAWs, of arbitrary length, starting at the origin and ending at the fixed site x ($\neq 0$). The ensemble is as in the free-endpoint case, with c_N replaced by $Nc_N(x)$. The connective constant μ and the critical exponent α_{sing} can be estimated from the Monte Carlo data, using the method of maximum likelihood.

A dynamic Monte Carlo algorithm for this ensemble was proposed by Berg and Foerster [79] and Aragão de Carvalho, Caracciolo and Fröhlich (BFACF) [75, 80]. The elementary moves are local deformations of the chain, with $\Delta N = 0, \pm 2$. The critical slowing-down in the BFACF algorithm is quite subtle. On the one hand, Sokal and Thomas [81] have proven the surprising result that $\tau_{exp} = +\infty$ for *all* $\beta \neq 0$ (see Section 8). On the other hand, numerical experiments [12, 82] show that $\tau_{int,N} \sim \langle N \rangle^{3.0 \pm 0.4}$ (in $d = 2$). Clearly, the BFACF dynamics is not well understood at present: further work, both theoretical and numerical, is needed.

In addition, Caracciolo, Pelissetto and Sokal [82] are studying a "hybrid" algorithm that combines local (BFACF) moves with non-local (cut-and-paste) moves. The critical slowing-down, measured in CPU units, appears to be reduced slightly compared to the pure BFACF algorithm: $\tau \sim \langle N \rangle^{\approx 2.3}$ in $d = 2$.

Canonical ensemble. Algorithms for this ensemble, based on local deformations of the chain, have been used by polymer physicists for more than 25 years [83, 84]. So the recent proof [85] that all such algorithms are nonergodic (= not irreducible) comes as a

[36] All these proofs are discussed in Section 8.

slight embarrassment. Fortunately, there does exist a *non-local* fixed-N algorithm which is ergodic: the "pivot" algorithm, invented by Lal [86] and independently reinvented by MacDonald *et al.* [87] and by Madras [16]. The elementary move is as follows: choose at random a pivot point k along the walk ($1 \le k \le N-1$); choose at random a non-identity element of the symmetry group of the lattice (rotation or reflection); then apply the symmetry-group element to $\omega_{k+1}, \ldots, \omega_N$ using ω_k as a pivot. The resulting walk is accepted if it is self-avoiding; otherwise it is rejected and the walk ω is counted once more in the sample. It can be proven [16] that this algorithm is ergodic and preserves the equal-weight probability distribution.

At first thought the pivot algorithm sounds terrible (at least it did to me): for N large, nearly all the proposed moves will get rejected. This is in fact true: the acceptance fraction behaves N^{-p} as $N \to \infty$, where $p \approx 0.19$ in $d = 2$ [16]. On the other hand, the pivot moves are very radical: after very few (5 or 10) accepted moves the SAW will have reached an "essentially new" conformation. One conjectures, therefore, that $\tau \sim N^p$. Actually it is necessary to be a bit more careful: for *global* observables f (such as the end-to-end distance ω_N^2) one expects $\tau_{int,f} \sim N^p$; but *local* observables (such as the angle between the 17^{th} and 18^{th} bonds of the walk) are expected to evolve a factor of N more slowly: $\tau_{int,f} \sim N^{1+p}$. Thus, the *slowest* mode is expected to behave as $\tau_{exp} \sim N^{1+p}$. For the pivot algorithm applied to *ordinary* random walk one can calculate the dynamical behavior exactly [16]: for *global* observables f the autocorrelation function behaves roughly like

$$\rho_{ff}(t) \;\sim\; \sum_{i=1}^{n} \left(1 - \frac{i}{n}\right)^t , \tag{7.12}$$

from which it follows that

$$\begin{aligned} \tau_{int,f} &\sim \log N & (7.13)\\ \tau_{exp,f} &\sim N & (7.14) \end{aligned}$$

— in agreement with our heuristic argument modulo logarithms. For the SAW, it is found numerically [16] that $\tau_{int,f} \sim N^{\approx 0.20}$ in $d = 2$, also in close agreement with the heuristic argument.

A careful analysis of the computational complexity of the pivot algorithm [36] shows that one "effectively independent" sample (at least as regards *global* observables) can be produced in a computer time of order N. This is a factor N more efficient than the Berretti-Sokal algorithm, a fact which opens up exciting prospects for high-precision Monte Carlo studies of critical phenomena in the SAW. Thus, with a modest computational effort (300 hours on a Cyber 170-730), Madras and I found $\nu = 0.7496 \pm 0.0007$ (95% confidence limits) for 2-dimensional SAWs of lengths $200 \le N \le 10000$ [16]. We hope to carry out soon a convincing numerical test of hyperscaling in the three-dimensional SAW.[37]

[37] **Note Added 1996:** This study has now been carried out [129], and yields $\nu = 0.5877 \pm 0.0006$ (subjective 68% confidence limits), based on SAWs of length up to 80000 steps. Proper treatment

8 Rigorous Analysis of Dynamic Monte Carlo Algorithms

In this final lecture, we discuss techniques for proving rigorous lower and upper bounds on the autocorrelation times of dynamic Monte Carlo algorithms. This topic is of primary interest, of course, to mathematical physicists: it constitutes the first steps toward a rigorous theory of dynamic critical phenomena, along lines parallel to the well-established rigorous theory of static critical phenomena. But these proofs are, I believe, also of some importance for practical Monte Carlo work, as they give insight into the physical basis of critical slowing-down and may point towards improved algorithms with reduced critical slowing-down.

There is a big difference between the techniques used for proving lower and upper bounds, and it is easy to understand this physically. To prove a *lower* bound on the autocorrelation time, it suffices to find *one* physical reason why the dynamics should be slow, i.e. to find *one* "slow mode". This physical insight can often be converted directly into a rigorous proof, using the variational method described below. On the other hand, to prove an *upper* bound on the autocorrelation time, it is necessary to understand *all* conceivable physical reasons for slowness, and to prove that *none* of them cause too great a slowing-down. This is extremely difficult to do, and has been carried to completion in very few cases.

We shall be obliged to restrict attention to *reversible* Markov chains [i.e. those satisfying the detailed-balance condition (2.27)], as these give rise to self-adjoint operators. Non-reversible Markov chains, corresponding to non-self-adjoint operators, are much more difficult to analyze rigorously.

The principal method for proving lower bounds on τ_{exp} (and in fact on $\tau_{int,f}$) is the *variational* (or *Rayleigh-Ritz*) *method.* Let us recall some of the theory of reversible Markov chains from Section 2: Let π be a probability measure, and let $P = \{p_{xy}\}$ be a transition probability matrix that satisfies detailed balance with respect to π. Now P acts naturally on functions (observables) according to

$$(Pf)(x) \;\equiv\; \sum_y p_{xy}\, f(y)\,. \tag{8.1}$$

In particular, when acting on the space $l^2(\pi)$ of π-square-integrable functions, the operator P is a *self-adjoint contraction.* Its spectrum therefore lies in the interval $[-1, 1]$. Moreover, P has an eigenvalue 1 with eigenvector equal to the constant function $\mathbf{1}$. Let Π be the orthogonal projector in $l^2(\pi)$ onto the constant functions. Then, for each real-valued observable $f \in l^2(\pi)$, we have

$$C_{ff}(t) \;=\; (f,\, (P^{|t|} - \Pi)f)$$

---

of corrections to scaling is crucial in obtaining this estimate. We also show that the interpenetration ratio Ψ approaches its limiting value $\Psi^* = 0.2471 \pm 0.0003$ *from above*, contrary to the prediction of the two-parameter renormalization-group theory. We have critically reexamined this theory and shown where the error lies [130, 129].

$$= \; (f,\, (I-\Pi) P^{|t|} (I-\Pi) f) \tag{8.2}$$

where $(g,h) \equiv \langle g^* \, h \rangle_\pi$ denotes the inner product in $l^2(\pi)$. By the spectral theorem, this can be written as

$$C_{ff}(t) \;=\; \int_{-1}^{1} \lambda^{|t|} \, d\nu_{ff}(\lambda)\,, \tag{8.3}$$

where $d\nu_{ff}$ is a *positive* measure. It follows that

$$\begin{aligned} \tau_{int,f} \;&=\; \frac{1}{2} \frac{\int_{-1}^{1} \frac{1+\lambda}{1-\lambda}\, d\nu_{ff}(\lambda)}{\int_{-1}^{1} d\nu_{ff}(\lambda)} \\ &\geq\; \frac{1}{2} \frac{1+\rho_{ff}(1)}{1-\rho_{ff}(1)} \end{aligned} \tag{8.4}$$

by Jensen's inequality (since the function $\lambda \mapsto (1+\lambda)/(1-\lambda)$ is convex).[38]

Our method will be to compute explicitly a lower bound on the normalized autocorrelation function at time lag 1,

$$\rho_{ff}(1) \;=\; \frac{C_{ff}(1)}{C_{ff}(0)}\,, \tag{8.5}$$

for a suitably chosen trial observable f. Equivalently, we compute an upper bound on the Rayleigh quotient

$$\frac{(f,\,(I-P)f)}{(f,\,(I-\Pi)f)} \;=\; \frac{C_{ff}(0) - C_{ff}(1)}{C_{ff}(0)} \;=\; 1 - \rho_{ff}(1)\,. \tag{8.6}$$

The crux of the matter is to pick a trial observable f that has a strong enough overlap with the "slowest mode". A useful formula is

$$(f,\,(I-P)f) \;=\; \frac{1}{2} \sum_{x,y} \pi_x \, p_{xy} \, |f(x) - f(y)|^2\,. \tag{8.7}$$

That is, the numerator of the Rayleigh quotient is half the mean-square change in f in a single time step.

Example 1. *Single-site heat-bath algorithm.* Let us consider, for simplicity, a translation-invariant spin model in a periodic box of volume V. Let $P_i = \{P_i(\varphi \to \varphi')\}$ be the transition probability matrix associated with applying the heat-bath operation at site i.[39] Then the operator P_i has the following properties:

[38] It also follows from (8.3) that $\rho_{ff}(t) \geq \rho_{ff}(1)^{|t|}$ for *even* values of t. Moreover, this holds for odd values of t if $d\nu_{ff}$ is supported on $\lambda \geq 0$ (though not necessarily otherwise); this is the case for the heat-bath and Swendsen-Wang algorithms, in which $P \geq 0$. Therefore, the decay as $t \to \infty$ of $\rho_{ff}(t)$ is also bounded below in terms of $\rho_{ff}(1)$.

[39] Probabilistically, P_i is the conditional expectation $E^\pi(\,\cdot \mid \{\varphi_j\}_{j\neq i})$. Analytically, P_i is the orthogonal projection in $l^2(\pi)$ onto the linear subspace consisting of functions depending only on $\{\varphi_j\}_{j\neq i}$.

(a) P_i is an orthogonal projection. (In particular, $P_i^2 = P_i$ and $0 \le P_i \le I$.)

(b) $P_i \mathbf{1} = \mathbf{1}$. (In particular, $\Pi P_i = P_i \Pi = \Pi$.)

(c) $P_i f = f$ if f depends only on $\{\varphi_j\}_{j \neq i}$. (In fact a stronger property holds: $P_i(fg) = f P_i g$ if f depends only on $\{\varphi_j\}_{j\neq i}$. But we shall not need this.)

For simplicity we consider only random site updating. Then the transition matrix of the heat-bath algorithm is

$$P \;=\; \frac{1}{V} \sum_i P_i \,. \tag{8.8}$$

We shall use the notation of the Ising model, but the same proof applies to much more general models.

As explained in Section 4, the critical slowing-down of the local algorithms (such as single-site heat-bath) arises from the fact that large regions in which the net magnetization is positive or negative tend to move slowly. In particular, the total magnetization of the lattice fluctuates slowly. So it is natural to expect that that the total magnetization $\mathcal{M} = \sum_i \sigma_i$ will be a "slow mode". Indeed, let us compute an upper bound on the Rayleigh quotient (8.6) with $f = \mathcal{M}$. The denominator is

$$(\mathcal{M},\, (I-\Pi)\mathcal{M}) \;=\; \langle \mathcal{M}^2 \rangle - \langle \mathcal{M} \rangle^2 \;=\; V\chi \tag{8.9}$$

(since $\langle \mathcal{M} \rangle = 0$ by symmetry). The numerator is

$$\begin{aligned}
(\mathcal{M},\, (I-P)\mathcal{M}) \;&=\; \frac{1}{V} \sum_{i,j,k} (\sigma_i,\, (I-P_j)\sigma_k) \\
&=\; \frac{1}{V} \sum_i (\sigma_i,\, (I-P_i)\sigma_i) \\
&\le\; \frac{1}{V} \sum_i (\sigma_i,\, \sigma_i) \\
&=\; \frac{1}{V} \sum_i \langle \sigma_i^2 \rangle \\
&=\; 1\,.
\end{aligned} \tag{8.10}$$

Here we first used properties (a) and (c) to deduce that $(\sigma_i,\, (I-P_j)\sigma_k) = 0$ unless $i = j = k$, and then used property (a) to bound $(\sigma_i,\, (I-P_i)\sigma_i)$. It follows from (8.9) and (8.10) that

$$1 - \rho_{\mathcal{M},\mathcal{M}}(1) \;\le\; \frac{1}{V\chi} \tag{8.11}$$

and hence that

$$\tau_{int,\mathcal{M}} \;\ge\; V\chi - \frac{1}{2}\,. \tag{8.12}$$

Note, however, that here time is measured in *hits of a single site*. If we measure time in *hits per site*, then we conclude that

$$\tau_{int,\mathcal{M}}^{(HPS)} \;\geq\; \chi - \frac{1}{2V} \;\approx\; \chi\,. \tag{8.13}$$

This proves a lower bound [88, 89, 90] on the dynamic critical exponent z, namely $z \geq \gamma/\nu$. (Here γ and ν are the static critical exponents for the susceptibility and correlation length, respectively. Note that by the usual static scaling law, $\gamma/\nu = 2 - \eta$; and for most models η is very close to zero. So this argument almost makes rigorous (as a *lower bound*) the heuristic argument that $z \approx 2$.)

Virtually the same argument applies, in fact, to *any* reversible single-site algorithm (e.g. Metropolis). The only difference is that the property $0 \leq P_i \leq I$ is replaced by $-I \leq P_i \leq I$, so the bound on the numerator is a factor of 2 larger. Hence

$$\tau_{int,\mathcal{M}}^{(HPS)} \;\geq\; \frac{\chi}{2} - \frac{1}{2V} \;\approx\; \frac{\chi}{2}\,. \tag{8.14}$$

Example 2. *Swendsen-Wang algorithm.* Recall that the SW algorithm simulates the joint model (6.6) by alternately applying the conditional distributions of $\{n\}$ given $\{\sigma\}$, and $\{\sigma\}$ given $\{n\}$ — that is, by alternately generating new bond occupation variables (independent of the old ones) given the spins, and new spin variables (independent of the old ones) given the bonds. The transition matrix $P_{SW} = P_{SW}(\{\sigma, n\} \to \{\sigma', n'\})$ is therefore a product

$$P_{SW} \;=\; P_{bond} P_{spin}\;, \tag{8.15}$$

where P_{bond} is the conditional expectation operator $E(\,\cdot\,|\{\sigma\})$, and P_{spin} is $E(\,\cdot\,|\{n\})$.

We shall use the variational method, with f chosen to be the bond density

$$f \;=\; \mathcal{N} \;=\; \sum_{\langle ij\rangle} n_{ij}\;. \tag{8.16}$$

To lighten the notation, let us write $\delta_{\sigma_b} \equiv \delta_{\sigma_i,\sigma_j}$ for a bond $b = \langle ij\rangle$. We then have

$$E(n_b|\{\sigma\}) \;=\; p_b\,\delta_{\sigma_b} \tag{8.17}$$

$$E(n_b n_{b'}|\{\sigma\}) \;=\; p_b\,p_{b'}\,\delta_{\sigma_b}\,\delta_{\sigma_{b'}} \qquad \text{for } b \neq b' \tag{8.18}$$

It follows from (8.17) that

$$\begin{aligned}
\langle n_b(t=0)\,n_{b'}(t=1)\rangle \;&=\; \langle n_b\,E(E(n_{b'}|\{\sigma\})|\{n\})\rangle \\
&=\; \langle n_b\,E(n_{b'}|\{\sigma\})\rangle \\
&=\; \langle E(n_b|\{\sigma\})\,E(n_{b'}|\{\sigma\})\rangle \\
&=\; p_b\,p_{b'}\,\langle \delta_{\sigma_b}\,\delta_{\sigma_{b'}}\rangle\;.
\end{aligned} \tag{8.19}$$

The corresponding truncated (connected) correlation function is clearly

$$\langle n_b(t=0)\,;n_{b'}(t=1)\rangle \;=\; p_b\,p_{b'}\,\langle \delta_{\sigma_b}\,;\delta_{\sigma_{b'}}\rangle\;, \tag{8.20}$$

where $\langle A\,;B\rangle \equiv \langle AB\rangle - \langle A\rangle\langle B\rangle$. We have thus expressed a dynamic correlation function of the SW algorithm in terms of a static correlation function of the Potts model.

Now let us compute the same quantity at time lag 0: by (8.18), for $b \neq b'$ we have

$$\langle n_b(t=0)\,n_{b'}(t=0)\rangle \;=\; p_b\,p_{b'}\,\langle \delta_{\sigma_b}\,\delta_{\sigma_{b'}}\rangle \tag{8.21}$$

and hence

$$\langle n_b(t=0)\,;n_{b'}(t=0)\rangle \;=\; p_b\,p_{b'}\,\langle \delta_{\sigma_b}\,;\delta_{\sigma_{b'}}\rangle\;. \tag{8.22}$$

On the other hand, for $b = b'$ we clearly have

$$\begin{aligned}\langle n_b(t=0)\,;n_{b'}(t=0)\rangle \;&=\; \langle n_b\rangle - \langle n_b\rangle^2 \\ &=\; p_b\,\langle\delta_{\sigma_b}\rangle - p_b^2\,\langle\delta_{\sigma_b}\rangle^2\;.\end{aligned} \tag{8.23}$$

Consider now the usual case in which

$$p_b \;=\; \begin{cases} p & \text{for } b \in B \\ 0 & \text{otherwise}\end{cases} \tag{8.24}$$

for some family of bonds B. Combining (8.20), (8.22) and (8.23) and summing over b, b', we obtain

$$\langle \mathcal{N}(t=0);\mathcal{N}(t=1)\rangle \;=\; p^2\langle \mathcal{E}\,;\mathcal{E}\rangle \tag{8.25}$$

$$\langle \mathcal{N}(t=0);\mathcal{N}(t=0)\rangle \;=\; p^2\langle \mathcal{E}\,;\mathcal{E}\rangle - p(1-p)\langle\mathcal{E}\rangle \tag{8.26}$$

where $\mathcal{E} \equiv -\sum_{b\in B}\delta_{\sigma_b} \leq 0$ is the energy. Hence the normalized autocorrelation function at time lag 1 is exactly

$$\rho_{\mathcal{N}\mathcal{N}}(1) \;\equiv\; \frac{\langle \mathcal{N}(t=0);\mathcal{N}(t=1)\rangle}{\langle \mathcal{N}(t=0);\mathcal{N}(t=0)\rangle} \;=\; 1 - \frac{-(1-p)E}{pC_H - (1-p)E}\;, \tag{8.27}$$

where $E \equiv V^{-1}\langle\mathcal{E}\rangle$ is the mean energy and $C_H \equiv V^{-1}\langle\mathcal{E}\,;\mathcal{E}\rangle$ is the specific heat (V is the volume). At criticality, $p \to p_{crit} > 0$ and $E \to E_{crit} < 0$, so

$$\rho_{\mathcal{N}\mathcal{N}}(1) \;\geq\; 1 - \frac{\text{const}}{C_H}\;. \tag{8.28}$$

We now remark that although $P_{SW} = P_{bond}P_{spin}$ is not self-adjoint, the modified transition matrix $P'_{SW} \equiv P_{spin}P_{bond}P_{spin}$ *is* self-adjoint (and positive-semidefinite), and $\mathcal{N}$ has the same autocorrelation function for both:

$$(\mathcal{N}, (P_{bond}P_{spin})^t\,\mathcal{N}) \;=\; (\mathcal{N}, (P_{spin}P_{bond}P_{spin})^t\,\mathcal{N}) \tag{8.29}$$

It follows that

$$\rho_{\mathcal{N}\mathcal{N}}(t) \;\geq\; \rho_{\mathcal{N}\mathcal{N}}(1)^{|t|} \;\geq\; (1 - \mathrm{const}/C_H)^{|t|} \tag{8.30}$$

and hence

$$\tau_{int,\mathcal{N}} \;\geq\; \mathrm{const} \times C_H \;. \tag{8.31}$$

This proves a lower bound [62] on the dynamic critical exponent z_{SW}, namely $z_{SW} \geq \alpha/\nu$. (Here α and ν are the static critical exponents for the susceptibility and correlation length, respectively.) For the two-dimensional Potts models with $q = 2, 3, 4$, it is known exactly (but non-rigorously!) that $\alpha/\nu = 0, \frac{2}{5}, 1$, with multiplicative logarithmic corrections for $q = 4$ [91]. The bound on z_{SW} may be conceivably be sharp (or sharp up to a logarithm) for $q = 4$ [62].

Example 3. *Berretti-Sokal algorithm for SAWs.* Let us consider the observable

$$f(\omega) \;=\; |\omega| \; (= N) \,, \tag{8.32}$$

the total number of bonds in the walk. We have argued heuristically that $\tau \sim \langle N \rangle^2$; we can now make this argument rigorous as a *lower bound*. Indeed, it suffices to use (8.7): since the maximum change of $|\omega|$ in a single time step is ± 1, it follows that

$$(f, (I - P)f) \;\leq\; \frac{1}{2} \,. \tag{8.33}$$

On the other hand, the denominator of the Rayleigh quotient is

$$(f, (I - \Pi)f) \;=\; \langle N^2 \rangle - \langle N \rangle^2 \;. \tag{8.34}$$

Assuming the usual scaling behavior (7.1) for the c_N, we have

$$\langle N \rangle \;\approx\; \frac{\gamma}{1 - \beta\mu} \tag{8.35}$$

$$\langle N^2 \rangle \;\approx\; \frac{\gamma(\gamma+1)}{1 - \beta\mu} \tag{8.36}$$

asymptotically as $\beta \uparrow \beta_c = \mu^{-1}$, and hence

$$\tau_{int,N} \;\geq\; \mathrm{const} \times \langle N \rangle^2 \;. \tag{8.37}$$

A very different approach to proving lower bounds on the autocorrelation time τ_{exp} is the *minimum hitting-time argument* [81]. Consider a Markov chain with transition matrix P satisfying detailed balance for some probability measure π. If A and B are subsets of the state space S, let T_{AB} be the minimum time for getting from A to B with nonzero probability, i.e.

$$T_{AB} \;\equiv\; \min\{n\colon\; p_{xy}^{(n)} > 0 \text{ for some } x \in A,\; y \in B\} \tag{8.38}$$

Then the theorem asserts that if T_{AB} is large and this is not "justified" by the rarity of A and/or B in the equilibrium distribution π, then the autocorrelation time τ_{exp} must be large. More precisely:

Theorem 2 *Consider a Markov chain with transition matrix P satisfying detailed balance for the probability measure π. Let T_{AB} be defined as in (8.38). Then*

$$\tau_{exp} \;\geq\; \sup_{A,B \in S} \frac{2\,(T_{AB}-1)}{-\log\left(\pi(A)\,\pi(B)\right)} \tag{8.39}$$

Proof. Let $A, B \subset S$, and let $n < T_{AB}$. Then, by definition of T_{AB},

$$(\chi_A,\; P^n \chi_B)_{l^2(\pi)} \;\equiv\; \sum_{\substack{x\in A\\ y\in B}} \pi_x\, p_{xy}^{(n)} \;=\; 0\,. \tag{8.40}$$

On the other hand, $P\mathbf{1} = P^*\mathbf{1} = \mathbf{1}$. It follows that

$$(\chi_A - \pi(A)\mathbf{1},\; P^n(\chi_B - \pi(B)\mathbf{1})\,)_{l^2(\pi)} \;=\; -\pi(A)\pi(B)\,. \tag{8.41}$$

Now since P is a *self-adjoint* operator, we have

$$\|P^n \restriction \mathbf{1}^\perp\| \;=\; \|P \restriction \mathbf{1}^\perp\|^n \;=\; R^n\,, \tag{8.42}$$

where $R = e^{-1/\tau_{exp}}$ is the spectral radius (= norm) of $P \restriction \mathbf{1}^\perp$. Hence, by the Schwarz inequality,

$$\begin{aligned}
&\left|(\chi_A - \pi(A)\mathbf{1},\; P^n(\chi_B - \mathbf{1})\,)_{l^2(\pi)}\right| \\
&\quad\leq\; R^n\, \|\chi_A - \pi(A)\mathbf{1}\|_{l^2(\pi)}\, \|\chi_B - \pi(B)\mathbf{1}\|_{l^2(\pi)} \\
&\quad=\; R^n\, \pi(A)^{1/2}\,(1-\pi(A))^{1/2}\,\pi(B)^{1/2}\,(1-\pi(B))^{1/2} \\
&\quad\leq\; R^n\, \pi(A)^{1/2}\,\pi(B)^{1/2}
\end{aligned} \tag{8.43}$$

Combining (8.41) with (8.43) and taking $n = T_{AB} - 1$, we arrive after a little algebra at (8.39). □

Remark. Using Chebyshev polynomials, a stronger bound can be proven: roughly speaking, τ_{exp} is bounded below by the *square* of the RHS of (8.39). For details, see [81, Theorem 3.1].

Let us apply this theorem to the BFACF algorithm for variable-length fixed-endpoint SAWs. Let ω^* be a fixed short walk from 0 to x, and let ω^n be a quasi-rectangular walk from 0 to x of linear size $\approx n$. Then $\pi(\omega^*) \sim 1$ and $\pi(\omega^n) \sim \beta^{\approx 4n}$, so that $-\log(\pi(\omega^*)\pi(\omega^n)) \sim n$. On the other hand — and this is the key point — the minimum time required to get from ω^n to ω^* (or vice versa) in the BFACF algorithm is of order n^2, since the *surface area* spanned by $\omega^n \cup \omega^*$ can change by at most one unit in a local deformation. Applying the theorem with $A = \{\omega^n\}$ and $B = \{\omega^*\}$, we obtain

$$\tau_{exp} \;\geq\; \sup_n \frac{\sim n^2}{\sim n} \;=\; +\infty. \tag{8.44}$$

As noted at the beginning of this lecture, it is *much* more difficult to prove *upper* bounds on the autocorrelation time — or even to prove that $\tau_{exp} < \infty$ — and few nontrivial results have been obtained so far.

The only *general* method (to my knowledge) for proving upper bounds on the autocorrelation time is Cheeger's inequality [77, 92, 93], the basic idea of which is to search for "bottlenecks" in the state space.[40] Consider first the rate of probability flow, in the stationary Markov chain, from a set A to its complement A^c, normalized by the invariant probabilities of A and A^c:

$$k(A) \;\equiv\; \frac{\sum\limits_{x\in A,\, y\in A^c} \pi_x\, p_{xy}}{\pi(A)\,\pi(A^c)} \;=\; \frac{(\chi_A,\; P\chi_{A^c})_{l^2(\pi)}}{\pi(A)\,\pi(A^c)}\;. \tag{8.45}$$

Now look for the *worst* such decomposition into A and A^c:

$$k \;\equiv\; \inf_{A:\, 0<\pi(A)<1} k(A)\;. \tag{8.46}$$

If, for some set A, the flow from A to A^c is very small compared to the invariant probabilities of A and A^c, then intuitively the Markov chain must have very slow convergence to equilibrium (the sets A and A^c are "metastable"). For reversible Markov chains a trivial variational argument makes this intuition rigorous: just take $f = \chi_A$. What is much more exciting is the converse: if there does *not* exist a set A for which the flow from A to A^c is unduly small, then the Markov chain must have *rapid* convergence to equilibrium, in the sense that the modified autocorrelation time τ'_{exp} is small. The precise statement is the following [77, Theorem 2.1]:

Theorem 3 *Let P be a transition probability matrix satisfying detailed balance for π, and let k be defined as above. Then*

$$\frac{-1}{\log(1-k)} \;\le\; \tau'_{exp} \;\le\; \frac{-1}{\log(1-\frac{k^2}{8})}\;. \tag{8.47}$$

The proof is not difficult, but enough is enough — the interested reader can look it up in the original paper.

Since this is a *two-sided* bound, we see that τ'_{exp} is finite *if and only if* $k > 0$. So in principle Cheeger's inequality can be used to prove exponential convergence to equilibrium whenever it holds. But in practice it is almost impossible to control the infimum (8.46) over *all* sets A. The most tractable case seems to be when the state

[40]**Note Added 1996:** There is now an extensive literature (largely by probabilists and theoretical computer scientists) on upper bounds on the autocorrelation time for Markov chains, based on inequalities of Cheeger, Poincaré, Nash and log-Sobolev types. For reviews, see e.g. [131, 132, 133, 134, 135, 136].

space is a *tree*: then A can always be chosen so that it is connected to A^c by a single bond. Using this fact, Lawler and Sokal [77] used Cheeger's inequality to prove

$$\tau'_{exp} \le \text{const} \times \langle N \rangle^{2\gamma} \tag{8.48}$$

for the Berretti-Sokal algorithm for SAWs.

A very different proof of an upper bound on τ'_{exp} in the Berretti-Sokal algorithm was given by Sokal and Thomas [10]. Their method is to study in detail the exponential moments of the hitting times from an arbitrary walk ω to the empty walk. Using a sequence of identities, they are able to write an algebraic inequality for such a moment in terms of itself; this inequality says roughly that the moment is either small or huge (where "huge" includes the possibility of $+\infty$) but cannot lie in-between (there is a "forbidden interval"). Then, by a continuity argument, they are able to rule out the possibility that the moment is huge. So it must be small. The final result is

$$\tau'_{exp} \le \text{const} \times \langle N \rangle^{1+\gamma}\,, \tag{8.49}$$

slightly better than the Lawler-Sokal bound.[41]

For spin models and lattice field theories, almost nothing is known about upper bounds on the autocorrelation time, except at high temperature. For the single-site heat-bath dynamics, it is easy to show that $\tau_{exp} < \infty$ (uniformly in the volume) above the Dobrushin uniqueness temperature; indeed, this is precisely what the standard proof of the Dobrushin uniqueness theorem [94, 95] does. One expects the same result to hold for *all* temperatures above critical, but this remains an open problem, despite recent progress by Aizenman and Holley [96].[42]

Acknowledgments

Many of the ideas reported here have grown out of joint work with my colleagues Alberto Berretti, Frank Brown, Sergio Caracciolo, Robert Edwards, Jonathan Goodman, Tony Guttmann, Greg Lawler, Xiao-Jian Li, Neal Madras, Andrea Pelissetto, Larry Thomas and Dan Zwanziger. I thank them for many pleasant and fruitful collaborations. In addition, I would like to thank Mal Kalos for introducing me to Monte Carlo work and for patiently answering my first naive questions.

The research of the author was supported in part by the U.S. National Science Foundation grants DMS–8705599 and DMS–8911273, the U.S. Department of Energy contract DE-FG02-90ER40581, the John Von Neumann National Supercomputer Center[†]

[41]**Note Added 1996:** See also [137], which recovers the bound (8.49) by a much simpler argument using the Poincaré inequality [131].

[42]**Note Added 1996:** There has been great progress on this problem in recent years: see [138, 139, 140, 141, 142, 143] and references cited therein.

[†]Deceased.

grant NAC–705, and the Pittsburgh Supercomputing Center grant PHY890035P. Acknowledgment is also made to the donors of the Petroleum Research Fund, administered by the American Chemical Society, for partial support of this research.

References

[1] N. Bakhvalov, *Méthodes Numériques* (MIR, Moscou, 1976), Chapter 5.

[2] J.M. Hammersley and D.C. Handscomb, *Monte Carlo Methods* (Methuen, London, 1964), Chapter 5.

[3] W.W. Wood and J.J. Erpenbeck, Ann. Rev. Phys. Chem. **27**, 319 (1976).

[4] J. G. Kemeny and J. L. Snell, *Finite Markov Chains* (Springer, New York, 1976).

[5] M. Iosifescu, *Finite Markov Processes and Their Applications* (Wiley, Chichester, 1980).

[6] K. L. Chung, *Markov Chains with Stationary Transition Probabilities*, 2^{nd} ed. (Springer, New York, 1967).

[7] E. Nummelin, *General Irreducible Markov Chains and Non-Negative Operators* (Cambridge Univ. Press, Cambridge, 1984).

[8] D. Revuz, *Markov Chains* (North-Holland, Amsterdam, 1975).

[9] Z. Šidak, Czechoslovak Math. J. **14**, 438 (1964).

[10] A. D. Sokal and L. E. Thomas, J. Stat. Phys. **54**, 797 (1989).

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

[12] S. Caracciolo and A.D. Sokal, J. Phys. **A19**, L797 (1986).

[13] D. Goldsman, Ph.D. thesis, School of Operations Research and Industrial Engineering, Cornell University (1984); L. Schruben, Oper. Res. **30**, 569 (1982) and **31**, 1090 (1983).

[14] M.B. Priestley, *Spectral Analysis and Time Series*, 2 vols. (Academic, London, 1981), Chapters 5-7.

[15] T.W. Anderson, *The Statistical Analysis of Time Series* (Wiley, New York, 1971).

[16] N. Madras and A.D. Sokal, J. Stat. Phys. **50**, 109 (1988).

[17] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[18] W.K. Hastings, Biometrika **57**, 97 (1970).

[19] J. Goodman and N. Madras, Lin. Alg. Appl. **216**, 61 (1995).

[20] J. Goodman and A.D. Sokal, Phys. Rev. **D40**, 2035 (1989).

[21] G.F. Mazenko and O.T. Valls, Phys. Rev. **B24**, 1419 (1981); C. Kalle, J. Phys. **A17**, L801 (1984); J.K. Williams, J. Phys. **A18**, 49 (1985); R.B. Pearson, J.L. Richardson and D. Toussaint, Phys. Rev. **B31**, 4472 (1985); S. Wansleben and D.P. Landau, J. Appl. Phys. **61**, 3968 (1987).

[22] M.N. Barber, in *Phase Transitions and Critical Phenomena*, vol. 8, ed. C. Domb and J.L. Lebowitz (Academic Press, London, 1983).

[23] K. Binder, J. Comput. Phys. **59**, 1 (1985).

[24] G. Parisi, in *Progress in Gauge Field Theory* (1983 Cargèse lectures), ed. G. 't Hooft *et al.* (Plenum, New York, 1984); G.G. Batrouni, G.R. Katz, A.S. Kronfeld, G.P. Lepage, B. Svetitsky and K.G. Wilson, Phys. Rev. **D32**, 2736 (1985); C. Davies, G. Batrouni, G. Katz, A. Kronfeld, P. Lepage, P. Rossi, B. Svetitsky and K. Wilson, J. Stat. Phys. **43**, 1073 (1986); J.B. Kogut, Nucl. Phys. **B275** [FS17], 1 (1986); S. Duane, R. Kenway, B.J. Pendleton and D. Roweth, Phys. Lett. **176B**, 143 (1986); E. Dagotto and J.B. Kogut, Phys. Rev. Lett. **58**, 299 (1987) and Nucl. Phys. **B290** [FS20], 451 (1987).

[25] J. Goodman and A.D. Sokal, Phys. Rev. Lett. **56**, 1015 (1986).

[26] R.G. Edwards, J. Goodman and A.D. Sokal, Nucl. Phys. **B354**, 289 (1991).

[27] R.H. Swendsen and J.-S. Wang, Phys. Rev. Lett. **58**, 86 (1987).

[28] R.P. Fedorenko, Zh. Vychisl. i Mat. Fiz. **4**, 559 (1964) [USSR Comput. Math. and Math. Phys. **4**, 227 (1964)].

[29] L.P. Kadanoff, Physics **2**, 263 (1966).

[30] K.G. Wilson, Phys. Rev. **B4**, 3174, 3184 (1971).

[31] A. Brandt, in *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics* (Paris, July 1972), ed. H. Cabannes and R. Temam (Lecture Notes in Physics #18) (Springer, Berlin, 1973); R.A. Nicolaides, J. Comput. Phys. **19**, 418 (1975) and Math. Comp. **31**, 892 (1977); A. Brandt, Math. Comp. **31**, 333 (1977); W. Hackbusch, in *Numerical Treatment of Differential Equations* (Oberwolfach, July 1976), ed. R. Bulirsch, R.D. Griegorieff and J. Schröder (Lecture Notes in Mathematics #631) (Springer, Berlin, 1978).

[32] W. Hackbusch, *Multi-Grid Methods and Applications* (Springer, Berlin, 1985).

[33] W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods* (Lecture Notes in Mathematics #960) (Springer, Berlin, 1982); Proceedings of the First Copper Mountain Conference on Multigrid Methods, ed. S. McCormick and U. Trottenberg, Appl. Math. Comput. **13**, no. 3-4, 213-470 (1983); D.J. Paddon and H. Holstein, editors, *Multigrid Methods for Integral and Differential Equations* (Clarendon Press, Oxford, 1985); Proceedings of the Second Copper Mountain Conference on Multigrid Methods, ed. S. McCormick, Appl. Math. Comput. **19**, no. 1-4, 1-372 (1986); W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods II* (Lecture Notes in Mathematics #1228) (Springer, Berlin, 1986); S.F. McCormick, editor, *Multigrid Methods* (SIAM, Philadelphia, 1987); *Proceedings of the Third Copper Mountain Conference on Multigrid Methods*, ed. S. McCormick (Dekker, New York, 1988).

[34] W.L. Briggs, *A Multigrid Tutorial* (SIAM, Philadelphia, 1987).

[35] R.S. Varga, *Matrix Iterative Analysis* (Prentice-Hall, Englewood Cliffs, N.J., 1962).

[36] H.R. Schwarz, H. Rutishauer and E. Stiefel, *Numerical Analysis of Symmetric Matrices* (Prentice-Hall, Englewood Cliffs, N.J., 1973).

[37] A.M. Ostrowski, Rendiconti Mat. e Appl. **13**, 140 (1954) at p. $141n$ [A. Ostrowski, *Collected Mathematical Papers*, vol. 1 (Birkhaüser, Basel, 1983), p. $169n$].

[38] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables* (Academic Press, New York–London, 1970).

[39] S.F. McCormick and J. Ruge, Math. Comp. **41**, 43 (1983).

[40] J. Mandel, S. McCormick and R. Bank, in *Multigrid Methods*, ed. S.F. McCormick (SIAM, Philadelphia, 1987), Chapter 5.

[41] J. Goodman and A.D. Sokal, unpublished.

[42] R.G. Edwards, S.J. Ferreira, J. Goodman and A.D. Sokal, Nucl. Phys. **B380**, 621 (1992).

[43] S.L. Adler, Phys. Rev. Lett. **60**, 1243 (1988).

[44] S.L. Adler, Phys. Rev. **D23**, 2091 (1981).

[45] C. Whitmer, Phys. Rev. **D29**, 306 (1984).

[46] E. Nelson, in *Constructive Quantum Field Theory*, ed. G. Velo and A. Wightman (Lecture Notes in Physics #25) (Springer, Berlin, 1973).

[47] B. Simon, *The $P(\varphi)_2$ Euclidean (Quantum) Field Theory* (Princeton Univ. Press, Princeton, N.J., 1974), Chapter I.

[48] R.B. Potts, Proc. Camb. Phil. Soc. **48**, 106 (1952).

[49] F.Y. Wu, Rev. Mod. Phys. **54**, 235 (1982); **55**, 315 (E) (1983).

[50] K. Krickeberg, *Probability Theory* (Addison-Wesley, Reading, Mass., 1965), Sections 4.2 and 4.5.

[51] R.H. Schonmann, J. Stat. Phys. **52**, 61 (1988).

[52] A.D. Sokal, unpublished.

[53] E.M. Reingold, J. Nievergelt and N. Deo, *Combinatorial Algorithms: Theory and Practice* (Prentice-Hall, Englewood Cliffs, N.J., 1977), Chapter 8; A. Gibbons, *Algorithmic Graph Theory* (Cambridge University Press, 1985), Chapter 1; S. Even, *Graph Algorithms* (Computer Science Press, Potomac, Maryland, 1979), Chapter 3.

[54] B.A. Galler and M.J. Fischer, Commun. ACM **7**, 301, 506 (1964); D.E. Knuth, *The Art of Computer Programming*, vol. 1, 2^{nd} ed., (Addison-Wesley, Reading, Massachusetts, 1973), pp. 353–355, 360, 572; J. Hoshen and R. Kopelman, Phys. Rev. **B14**, 3438 (1976).

[55] K. Binder and D. Stauffer, in *Applications of the Monte Carlo Method in Statistical Physics*, ed. K. Binder (Springer-Verlag, Berlin, 1984), section 8.4; D. Stauffer, *Introduction to Percolation Theory* (Taylor & Francis, London, 1985).

[56] Y. Shiloach and U. Vishkin, J. Algorithms **3**, 57 (1982).

[57] R.G. Edwards, X.-J. Li and A.D. Sokal, Sequential and vectorized algorithms for computing the connected components of an undirected graph, unpublished (and uncompleted) notes.

[58] P.W. Kasteleyn and C.M. Fortuin, J. Phys. Soc. Japan **26** (Suppl.), 11 (1969); C.M. Fortuin and P.W. Kasteleyn, Physica **57**, 536 (1972); C.M. Fortuin, Physica **58**, 393 (1972); C.M. Fortuin, Physica **59**, 545 (1972).

[59] C.-K. Hu, Phys. Rev. **B29**, 5103 (1984); T.A. Larsson, J. Phys. **A19**, 2383 (1986) and **A20**, 2239 (1987).

[60] R.G. Edwards and A.D. Sokal, Phys. Rev. **D38**, 2009 (1988).

[61] A.D. Sokal, in *Computer Simulation Studies in Condensed Matter Physics: Recent Developments* (Athens, Georgia, 15–26 February 1988), ed. D.P. Landau, K.K. Mon and H.-B. Schüttler (Springer-Verlag, Berlin–Heidelberg, 1988).

[62] X.-J. Li and A.D. Sokal, Phys. Rev. Lett. **63**, 827 (1989).

[63] W. Klein, T. Ray and P. Tamayo, Phys. Rev. Lett. **62**, 163 (1989).

[64] T.S. Ray, P. Tamayo and W. Klein, Phys. Rev. **A39**, 5949 (1989).

[65] U. Wolff, Phys. Rev. Lett. **62**, 361 (1989).

[66] U. Wolff, Phys. Lett. **B228**, 379 (1989); P. Tamayo, R.C. Brower and W. Klein, J. Stat. Phys. **58**, 1083 (1990) and **60**, 889 (1990).

[67] D. Kandel, E. Domany, D. Ron, A. Brandt and E. Loh, Phys. Rev. Lett. **60**, 1591 (1988); D. Kandel, E. Domany and A. Brandt, Phys. Rev. **B40**, 330 (1989).

[68] R.G. Edwards and A.D. Sokal, Swendsen-Wang algorithm augmented by duality transformations for the two-dimensional Potts model, in preparation.

[69] R. Ben-Av, D. Kandel, E. Katznelson, P.G. Lauwers and S. Solomon, J. Stat. Phys. **58**, 125 (1990); R.C. Brower and S. Huang, Phys. Rev. **B41**, 708 (1990).

[70] F. Niedermayer, Phys. Rev. Lett. **61**, 2026 (1988).

[71] R.C. Brower and P. Tamayo, Phys. Rev. Lett. **62**, 1087 (1989).

[72] U. Wolff, Nucl. Phys. **B322**, 759 (1989); Phys. Lett. **B222**, 473 (1989); Nucl. Phys. **B334**, 581 (1990).

[73] R.G. Edwards and A.D. Sokal, Phys. Rev. **D40**, 1374 (1989).

[74] P.G. DeGennes, *Scaling Concepts in Polymer Physics* (Cornell Univ. Press, Ithaca NY, 1979).

[75] C. Aragão de Carvalho, S. Caracciolo and J. Fröhlich, Nucl. Phys. **B215** [FS7], 209 (1983).

[76] A. Berretti and A.D. Sokal, J. Stat. Phys. **40**, 483 (1985).

[77] G. Lawler and A.D. Sokal, Trans. Amer. Math. Soc. **309**, 557 (1988).

[78] G. Lawler and A.D. Sokal, in preparation.

[79] B. Berg and D. Foerster, Phys. Lett. **106B**, 323 (1981).

[80] C. Aragão de Carvalho and S. Caracciolo, J. Physique **44**, 323 (1983).

[81] A. D. Sokal and L. E. Thomas, J. Stat. Phys. **51**, 907 (1988).

[82] S. Caracciolo, A. Pelissetto and A.D. Sokal, J. Stat. Phys. **60**, 1 (1990).

[83] M. Delbrück, in *Mathematical Problems in the Biological Sciences* (Proc. Symp. Appl. Math., vol. 14), ed. R.E. Bellman (American Math. Soc., Providence RI, 1962).

[84] P.H. Verdier and W.H. Stockmayer, J. Chem. Phys. **36**, 227 (1962).

[85] N. Madras and A.D. Sokal, J. Stat. Phys. **47**, 573 (1987).

[86] M. Lal, Molec. Phys. **17**, 57 (1969).

[87] B. MacDonald *et al.*, J. Phys. **A18**, 2627 (1985).

[88] K. Kawasaki, Phys. Rev. **148**, 375 (1966).

[89] B.I. Halperin, Phys. Rev. **B8**, 4437 (1973).

[90] R. Alicki, M. Fannes and A. Verbeure, J. Stat. Phys. **41**, 263 (1985).

[91] M.P.M. den Nijs, J. Phys. **A12**, 1857 (1979); J.L. Black and V.J. Emery, Phys. Rev. **B23**, 429 (1981); B. Nienhuis, J. Stat. Phys. **34**, 731 (1984).

[92] A. Sinclair and M. Jerrum, Information and Computation **82**, 93 (1989).

[93] R. Brooks, P. Doyle and R. Kohn, in preparation.

[94] L.N. Vasershtein, Prob. Inform. Transm. **5**, 64 (1969).

[95] O.E. Lanford III, in *Statistical Mechanics and Mathematical Problems* (Lecture Notes in Physics #20), ed. A. Lenard (Springer, Berlin, 1973).

[96] M. Aizenman and R. Holley, in *Percolation Theory and Ergodic Theory of Infinite Particle Systems*, ed. H. Kesten (Springer, New York, 1987).

ADDED BIBLIOGRAPHY (DECEMBER 1996):

[97] M. Lüscher, P. Weisz and U. Wolff, Nucl. Phys. **B359**, 221 (1991).

[98] J.-K. Kim, Phys. Rev. Lett. **70**, 1735 (1993); Nucl. Phys. B (Proc. Suppl.) **34**, 702 (1994); Phys. Rev. **D50**, 4663 (1994); Europhys. Lett. **28**, 211 (1994); Phys. Lett. **B345**, 469 (1995).

[99] S. Caracciolo, R.G. Edwards, S.J. Ferreira, A. Pelissetto and A.D. Sokal, Phys. Rev. Lett. **74**, 2969 (1995).

[100] S.J. Ferreira and A.D. Sokal, Phys. Rev. **B51**, 6720 (1995).

[101] S. Caracciolo, R.G. Edwards, A. Pelissetto and A.D. Sokal, Phys. Rev. Lett. **75**, 1891 (1995).

[102] G. Mana, A. Pelissetto and A.D. Sokal, Phys. Rev. **D54**, R1252 (1996).

[103] G. Mana, A. Pelissetto and A.D. Sokal, `hep-lat/9610021`, submitted to Phys. Rev. D.

[104] C.-R. Hwang and S.-J. Sheu, in *Stochastic Models, Statistical Methods, and Algorithms in Image Analysis* (Lecture Notes in Statistics #74), ed. P. Barone, A. Frigessi and M. Piccioni (Springer, Berlin, 1992).

[105] B. Gidas, private communication (1991).

[106] T. Mendes and A.D. Sokal, Phys. Rev. **D53**, 3438 (1996).

[107] G. Mana, T. Mendes, A. Pelissetto and A.D. Sokal, Nucl. Phys. B (Proc. Suppl.) **47**, 796 (1996).

[108] T. Mendes and A. Pelissetto and A.D. Sokal, Nucl. Phys. **B477**, 203 (1996).

[109] A.D. Sokal, Nucl. Phys. B (Proc. Suppl.) **20**, 55 (1991).

[110] J.-S. Wang and R.H. Swendsen, Physica **A167**, 565 (1990).

[111] C.F. Baillie and P.D. Coddington, Phys. Rev. Lett. **68**, 962 (1992); and private communication.

[112] J. Salas and A.D. Sokal, J. Stat. Phys. **85**, 297 (1996).

[113] J. Salas and A.D. Sokal, Dynamic critical behavior of the Swendsen–Wang algorithm: The two-dimensional 3-state Potts model revisited, `hep-lat/9605018`, J. Stat. Phys. **87**, no. 1/2 (April 1997).

[114] J. Salas and A.D. Sokal, Logarithmic corrections and finite-size scaling in the two-dimensional 4-State Potts model, `hep-lat/9607030`, submitted to J. Stat. Phys.

[115] G.M. Torrie and J.P. Valleau, Chem. Phys. Lett. **28**, 578 (1974); J. Comput. Phys. **23**, 187 (1977); J. Chem. Phys. **66**, 1402 (1977).

[116] B.A. Berg and T. Neuhaus, Phys. Lett. **B267**, 249 (1991); Phys. Rev. Lett. **68**, 9 (1992).

[117] B.A. Berg, Int. J. Mod. Phys. **C4**, 249 (1993).

[118] A.D. Kennedy, Nucl. Phys. B (Proc. Suppl.) **30**, 96 (1993).

[119] W. Janke, in *Computer Simulations in Condensed Matter Physics VII*, eds. D.P. Landau, K.K. Mon and H.-B. Schüttler (Springer-Verlag, Berlin, 1994).

[120] S. Wiseman and E. Domany, Phys. Rev. E **48**, 4080 (1993).

[121] D. Kandel, R. Ben-Av and E. Domany, Phys. Rev. Lett. **65**, 941 (1990) and Phys. Rev. **B45**, 4700 (1992).

[122] J.-S. Wang, R.H. Swendsen and R. Kotecký, Phys. Rev. Lett. **63**, 109 (1989).

[123] J.-S. Wang, R.H. Swendsen and R. Kotecký, Phys. Rev. **B42**, 2465 (1990).

[124] M. Lubin and A.D. Sokal, Phys. Rev. Lett. **71**, 1778 (1993).

[125] X.-J. Li and A.D. Sokal, Phys. Rev. Lett. **67**, 1482 (1991).

[126] S. Caracciolo, R.G. Edwards, A. Pelissetto and A.D. Sokal, Nucl. Phys. **B403**, 475 (1993).

[127] A.D. Sokal, in *Monte Carlo and Molecular Dynamics Simulations in Polymer Science*, ed. K. Binder (Oxford University Press, New York, 1995).

[128] A.D. Sokal, Nucl. Phys. B (Proc. Suppl.) **47**, 172 (1996).

[129] B. Li, N. Madras and A.D. Sokal, J. Stat. Phys. **80**, 661 (1995).

[130] A.D. Sokal, Europhys. Lett. **27**, 661 (1994).

[131] P. Diaconis and D. Stroock, Ann. Appl. Prob. **1**, 36 (1991).

[132] A.J. Sinclair, *Randomised Algorithms for Counting and Generating Combinatorial Structures* (Birkhäuser, Boston, 1993).

[133] P. Diaconis and L. Saloff-Coste, Ann. Appl. Prob. **3**, 696 (1993).

[134] P. Diaconis and L. Saloff-Coste, Ann. Appl. Prob. **6**, 695 (1996).

[135] P. Diaconis and L. Saloff-Coste, J. Theoret. Prob. **9**, 459 (1996).

[136] M. Jerrum and A. Sinclair, in *Approximation Algorithms for NP-Hard Problems*, ed. D.S. Hochbaum (PWS Publishing, Boston, 1996).

[137] D. Randall and A.J. Sinclair, in *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms* (ACM Press, New York, 1994).

[138] D.W. Stroock and B. Zegarlinski, J. Funct. Anal. **104**, 299 (1992).

[139] D.W. Stroock and B. Zegarlinski, Commun. Math. Phys. **144**, 303 (1992).

[140] D.W. Stroock and B. Zegarlinski, Commun. Math. Phys. **149**, 175 (1992).

[141] S.L. Lu and H.T. Yau, Commun. Math. Phys. **156**, 399 (1993).

[142] F. Martinelli and E. Olivieri, Commun. Math. Phys. **161**, 447, 487 (1994).

[143] D.W. Stroock and B. Zegarlinski, J. Stat. Phys. **81**, 1007 (1995).