Journal of Statistical Physics, Vol. 60, Nos. 1/2, 1990

Nonlocal Monte Carlo Algorithm for Self-Avoiding Walks with Fixed Endpoints

Sergio Caracciolo,[1] Andrea Pelissetto,[1] and Alan D. Sokal[2]

Received October 17, 1989; revision received November 20, 1989

We study a new Monte Carlo algorithm for generating self-avoiding walks with variable length (controlled by a fugacity β) and fixed endpoints. The algorithm is a hybrid of local (BFACF) and nonlocal (cut-and-paste) moves. We find that the critical slowing-down, measured in units of computer time, is reduced compared to the pure BFACF algorithm: $\tau_{CPU} \sim \langle N \rangle^{\approx 2.3}$ versus $\langle N \rangle^{\approx 3.0}$. We also prove some rigorous bounds on the autocorrelation time for these and related Monte Carlo algorithms.

KEY WORDS: Self-avoiding walk; polymer; Monte Carlo; pivot algorithm; BFACF algorithm; cut-and-paste; critical exponent.

1. INTRODUCTION

The self-avoiding walk (SAW) is a well-known lattice model of a polymer molecule with excluded volume.[1,2] Its equivalence to the $n = 0$ limit of the n-vector model[3-8] has also made it an important test case in the theory of critical phenomena.

In this paper we study a new Monte Carlo algorithm for generating an ensemble of SAWs with *variable length* (controlled by a fugacity β) and *fixed endpoints*. This ensemble, which has been used in refs. 7 and 9–13, is the appropriate one for determining the critical exponent α_{sing}, which governs the critical singularities of the SAW analogue of the specific heat (see Section 2.1 for a precise definition). In particular, we can test the hyperscaling relation $d\nu = 2 - \alpha_{sing}$,[14-20] along the lines proposed in ref. 11.

Our algorithm is an extension of an earlier algorithm due to Berg and Foerster[21] and Aragão de Carvalho, Caracciolo, and Fröhlich,[7,9] here-

[1] Scuola Normale Superiore and INFN-Sezione di Pisa, Pisa 56100, Italy.
[2] Department of Physics, New York University, New York, New York 10003.

0022-4715/90/0700-0001$06.00/0

822/60/1-2-1

after called BFACF. The BFACF algorithm simulates an ensemble of variable-length SAWs with fixed endpoints, in which a SAW of N steps gets relative weight $N\beta^N$. The elementary moves of the BFACF algorithm are *local deformations* of the SAW, which amount to inserting or removing an elementary plaquette somewhere along the walk (hence $\Delta N = 0, \pm 2$ on cubic lattices). This algorithm has, however, a rather peculiar dynamical behavior characterized by the metastability of large quasirectangular configurations.[22,10] The resulting long autocorrelation time (severe critical slowing-down, $\tau_{\mathrm{int},N} \sim \langle N \rangle^{\approx 3}$) makes it difficult to obtain the high-precision estimates of critical exponents that are needed for a meaningful test of hyperscaling.

Our generalization of the BFACF algorithm is inspired by recent work by Madras and Sokal[23] on the pivot algorithm,[24,25],[3] which is a method for generating walks of fixed length and free endpoints by means of nonlocal ("pivot") moves. The moral of ref. 23 is that certain types of radically nonlocal moves can lead to extraordinarily efficient Monte Carlo algorithms, if the acceptance fraction for these moves is not too small (e.g., only a small inverse power of N) and the benefit from successful moves is sufficiently great. Madras and Sokal (ref. 23, Section 5.4) suggested the possibility of augmenting the BFACF algorithm by means of nonlocal "cut-and-paste" moves, the aim of which is to speed up equilibration within each subspace of fixed N (and in particular to destabilize the formerly metastable configurations).[4] This algorithm is thus a hybrid in which the nonlocal moves hopefully assure the rapid equilibration *within* subspaces of fixed N, while the local (BFACF) moves assure equilibration *between* different N (and in particular make the algorithm ergodic). The algorithm has a free parameter p_{nl}—the percentage of nonlocal moves—which can be tuned as a function of $\langle N \rangle$ to optimize the computational efficiency.

In the best case one can expect that the hybrid algorithm will achieve an autocorrelation time $\tau \sim \langle N \rangle^2$: for even if the nonlocal moves were to cause instant equilibration at fixed N, the local moves would still have to carry out a random walk in N. Such a behavior, if achieved, would be a significant improvement over the pure BFACF algorithm. This estimate

[3] We have recently discovered the following references, which should be added to those cited in ref. 23: The continuum version of the pivot algorithm was independently reinvented in 1974 by Curro[26]; it was subsequently used by Scott[27] and probably by others.

[4] Similar cut-and-paste moves have been employed very recently by Dubins *et al.*,[28] Madras *et al.*,[29] and Janse van Rensburg *et al.*,[30] in the context of a fixed-N, fixed-endpoint algorithm. Also, vaguely related methods have been employed by Olaj *et al.*,[31] Mansfield,[32] Madden,[33] and Reiter *et al.*[34] for multichain polymer systems, and by Pollock and Ceperley[35] for quantum Monte Carlo. We thank Marvin Bishop and Bernard Piller for bringing these latter references to our attention.

refers, however, to *physical* time units; since the nonlocal moves require a computer time that grows as a fractional power of $\langle N \rangle$, it is a subtle matter to choose p_{nl} so as to minimize the autocorrelation time as measured in *computer* (CPU) time units.

In this paper we carry out a detailed numerical and theoretical study of the BFACF/cut-and-paste algorithm, with emphasis on its dynamic critical behavior and on the problem of optimization.[5] Along the way we obtain a number of interesting general theorems concerning the behavior of "hybrid" Monte Carlo algorithms.

Our numerical experiments show that the *physical* autocorrelation time indeed scales as $\tau \sim \langle N \rangle^2$ at fixed p_{nl}. Taking into account the CPU time, we find that the optimal p_{nl} scales as $\sim 1/\langle N \rangle^{\approx 0.8}$, and the autocorrelation time in CPU units then scales as $\tau_{CPU} \sim \langle N \rangle^{\approx 2.3}$. For example, already at $\langle N \rangle \approx 100$ we find that the physical (resp. CPU) autocorrelation time of the hybrid algorithm with $p_{nl} = 0.05$ is a factor 6 (resp. 4) smaller than that of the pure BFACF algorithm. The hybrid algorithm provides, therefore, a substantial improvement over previous algorithms for fixed-endpoint SAWs.

The plan of this paper is as follows: In Section 2 we give a brief review of the self-avoiding walk (SAW) and dynamic Monte Carlo methods, and set the notation. In Section 3 we define the BFACF algorithm and discuss its dynamical behavior, define the cut-and-paste moves and discuss their expected effect, discuss the data structures needed for implementing the algorithm, and analyze the computational complexity. In Section 4 we present our numerical results. Section 5 contains some brief conclusions. In the Appendix we prove some general theorems about the dynamic behavior of reversible Markov chains; these theorems are likely to have other applications to Monte Carlo methods in statistical mechanics.

2. BACKGROUND AND NOTATION

2.1. The Self-Avoiding Walk (SAW)

In this section we review briefly the basic facts and conjectures about the SAW that will be used in the remainder of the paper. Let $\mathscr{L}$ be some regular d-dimensional lattice. Then an *N-step self-avoiding walk* (SAW) ω on $\mathscr{L}$ is a sequence of *distinct* points $\omega_0, \omega_1, \ldots, \omega_N$ in $\mathscr{L}$ such that each point is a nearest neighbor of its predecessor. For simplicity we shall restrict attention to the simple (hyper)cubic lattice $\mathbf{Z}^d$; similar ideas would

[5] A preliminary version of this work was reported at the Lattice '88 conference.[36]

apply to other regular lattices. We assume all walks to begin at the origin ($\omega_0 = 0$) unless stated otherwise.

Let $\mathscr{S}_N$ be the set of N-step SAWs on $\mathbf{Z}^d$ starting at the origin and ending anywhere, and let c_N be the cardinality of $\mathscr{S}_N$. Then it can be proven[37–39] that

$$\mu^N \leqslant c_N \leqslant K_1 \mu^N \exp(K_2 \sqrt{N}) \tag{2.1}$$

for suitable constants μ, K_1, and K_2.[6] Here μ is called the *connective constant* of the lattice, and it is easy to prove that $d \leqslant \mu \leqslant 2d-1$. It is strongly believed, though not yet proven, that c_N has the asymptotic behavior

$$c_N \sim \mu^N N^{\gamma-1} \tag{2.2}$$

as $N \to \infty$.[7] Here γ is a *critical exponent*, which is believed to be universal among lattices of a given dimension d. If (2.2) holds, then (2.1) implies that $\gamma \geqslant 1$.

Similarly, let $\mathscr{S}_N(x)$ be the set of N-step SAWs on $\mathbf{Z}^d$ starting at the origin and ending at x, and let $c_N(x)$ be the cardinality of $\mathscr{S}_N(x)$. Then it can be proven that[40,41,43]

$$K_3(x)\mu^N \exp(-K_4\sqrt{N}) \leqslant c_N(x) \leqslant K_5 \mu^N \exp(K_6\sqrt{N}) \tag{2.3}$$

for $x \neq 0$ and $N = (\sum_{i=1}^d x^{(i)}) \bmod 2$, with the *same* μ as in (2.1), for suitable constants $K_3(x)$, K_4, K_5, and K_6.[8] It is strongly believed, though not yet proven, that $c_N(x)$ has the asymptotic behavior

$$c_N(x) \sim \mu^N N^{\alpha_{\text{sing}}-2} \qquad (x \text{ fixed} \neq 0) \tag{2.4}$$

as $N \to \infty$, where α_{sing} is another (universal) critical exponent (independent of x).

Consider now the mean-square end-to-end distance

$$\langle \omega_N^2 \rangle \equiv \frac{1}{c_N} \sum_x |x|^2 c_N(x) \tag{2.5}$$

[6] A slightly stronger (for $d > 2$) upper bound on c_N has been proven by Kesten.[42]

[7] Very recently, Slade[46] has proven that (2.2) holds with $\gamma = 1$ for SAWs in sufficiently high dimension d.

[8] For $|x| = 1$, Hammersley[40] has proven that $c_N(x) \leqslant (N+1)\mu^N$, and Kesten[42] has proven two lower bounds on $c_N(x)$ that are slightly better than (2.3). Some stronger bounds have recently been proven by Madras.[44]

and the mean-square radii of gyration

$$\langle S_N^2 \rangle \equiv \frac{1}{c_N} \sum_{\omega \in \mathscr{S}_N} S_N^2(\omega) \tag{2.6}$$

$$\langle S_N^2 \rangle_x \equiv \frac{1}{c_N(x)} \sum_{\omega \in \mathscr{S}_N(x)} S_N^2(\omega) \tag{2.7}$$

where

$$\begin{aligned} S_N^2(\omega) &\equiv \frac{1}{N+1} \sum_{i=0}^{N} \left(\omega_i - \frac{1}{N+1} \sum_{j=0}^{N} \omega_j \right)^2 \\ &= \frac{1}{N+1} \sum_{i=0}^{N} \omega_i^2 - \left(\frac{1}{N+1} \sum_{i=0}^{N} \omega_i \right)^2 \end{aligned} \tag{2.8}$$

Very little has been proven rigorously about these quantities, but they are believed to have the asymptotic behavior

$$\langle \omega_N^2 \rangle \sim N^{2\nu} \tag{2.9}$$

$$\langle S_N^2 \rangle \sim N^{2\nu} \tag{2.10}$$

$$\langle S_N^2 \rangle_x \sim N^{2\nu} \qquad (x \text{ fixed} \neq 0) \tag{2.11}$$

as $N \to \infty$, where ν is another (universal) critical exponent.[9]

Finally, for walks $\omega \in \mathscr{S}_N(x)$ with $|x| = 1$ in *two dimensions*, we define $\mathscr{A}(\omega)$ to be the *signed* area enclosed by the closed loop $(\omega_0, \omega_1, \ldots, \omega_N, \omega_0)$, namely

$$\mathscr{A}(\omega) \equiv \text{“}\int y\, dx\text{”} \equiv \sum_{i=1}^{N+1} \omega_i^{(2)} (\omega_i^{(1)} - \omega_{i-1}^{(1)}) \tag{2.12}$$

where $\omega_{N+1} \equiv \omega_0$ and the superscripts refer to the 1- and 2-components of vectors in $\mathbf{Z}^2$. Clearly, $\langle \mathscr{A} \rangle_x = 0$ by reflection symmetry. It is believed, but not proven, that $\langle |\mathscr{A}| \rangle_x$ has the asymptotic behavior

$$\langle |\mathscr{A}| \rangle_x \sim N^{2\nu} \tag{2.13}$$

as $N \to \infty$.

[9] Very recently, Slade[45–47] has proven that (2.9) and (2.10) hold with $\nu = 1/2$ for SAWs in sufficiently high dimension d.

The names of the critical exponents γ, α_{sing}, and ν are chosen by analogy with the corresponding exponents in ferromagnetic spin systems.[15,48] Indeed, the generating functions of self-avoiding walks,

$$\chi(\beta) \equiv \sum_{N=0}^{\infty} \beta^N c_N \tag{2.14}$$

$$G(x;\beta) \equiv \sum_{N=0}^{\infty} \beta^N c_N(x) \tag{2.15}$$

are *equal* to the susceptibility and spin–spin correlation function in the n-vector model analytically continued to $n=0$.[5–8] In particular, if x is a nearest neighbor of the origin, then $G(x;\beta)$ is essentially the energy E (up to an additive and multiplicative constant). Inserting (2.2) and (2.4) into (2.14)–(2.15), we obtain the leading behavior

$$\chi(\beta) \sim (\beta_c - \beta)^{-\gamma} \tag{2.16}$$

$$G(x;\beta) \sim (\beta_c - \beta)^{1-\alpha_{\text{sing}}} + \text{regular terms} \tag{2.17}$$

as β approaches the *critical point* $\beta_c \equiv 1/\mu$. Note, in particular, that α_{sing} is the exponent for the *singular part* of the specific heat $C_H \sim \partial E/\partial\beta$; the exponent for the full specific heat is $\alpha = \max(\alpha_{\text{sing}}, 0)$.

By analogy with the (conjectured) hyperscaling relation for the specific heat in spin models,[14–20] it is reasonable to conjecture that

$$d\nu = 2 - \alpha_{\text{sing}} \tag{2.18}$$

for self-avoiding walks. One of the main objectives of the present paper is to devise a Monte Carlo algorithm for fixed-endpoint SAWs that is efficient enough to allow a high-precision test of this hyperscaling conjecture. For further discussion, see ref. 11.

2.2. Dynamic Monte Carlo

In this section we review briefly the principles of dynamic Monte Carlo methods, and define some quantities (autocorrelation times) that will play an important role in the remainder of the paper.

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. Dynamic methods are those that generate a sequence of *correlated* samples from some stochastic process (usually a Markov process) having the desired probability distribution π as its unique equilibrium distribution.

For simplicity let us assume that the state space S is discrete (i.e., finite or countably infinite); this is the case in the applications studied in this paper. Consider a Markov chain with state space S and transition probability matrix $P = \{p(x \to y)\} = \{p_{xy}\}$ satisfying the following two conditions:

(A) For each pair $x, y \in S$, there exists an $n \geqslant 0$ for which $p_{xy}^{(n)} > 0$. Here $p_{xy}^{(n)}$ is the n-step transition probability from x to y. [This condition is called *irreducibility* (or *ergodicity*); it asserts that each state can eventually be reached from each other state.]

(B) For each $y \in S$,

$$\sum_{x \in S} \pi_x p_{xy} = \pi_y \tag{2.19}$$

[This condition asserts that π is a stationary distribution for the Markov chain $P = \{p_{xy}\}$.]

In this case it can be shown[49] that π is the *unique* stationary distribution for the Markov chain $P = \{p_{xy}\}$, and that the occupation-time distribution over long time intervals converges (with probability 1) to π, irrespective of the initial state of the system. If, in addition, P is *aperiodic* (this means that for each pair $x, y \in S$, $p_{xy}^{(n)} > 0$ for *all* sufficiently large n), then the probability distribution at any single time in the far future also converges to π, irrespective of the initial state—that is, $\lim_{n \to \infty} p_{xy}^{(n)} = \pi_y$ for all x.

Thus, simulation of the Markov chain P provides a legitimate Monte Carlo method for estimating averages with respect to π. However, since the successive states $X_0, X_1, \ldots$ of the Markov chain are in general highly correlated, the variance of estimates produced in this way may be much higher than in independent sampling. To make this precise, let $A = \{A(x)\}_{x \in S}$ 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, "thermalize" it for a very long time prior to observing the system). Then $\{A_t\} \equiv \{A(X_t)\}$ is a stationary stochastic process with mean

$$\mu_A \equiv \langle A_t \rangle = \sum_{x \in S} \pi_x A(x) \tag{2.20}$$

and *unnormalized autocorrelation function*[10]

$$\begin{aligned} C_{AA}(t) &\equiv \langle A_s A_{s+t} \rangle - \mu_A^2 \\ &= \sum_{x, y \in S} A(x)[\pi_x p_{xy}^{(|t|)} - \pi_x \pi_y] A(y) \end{aligned} \tag{2.21}$$

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

The *normalized autocorrelation function* is then

$$\rho_{AA}(t) \equiv C_{AA}(t)/C_{AA}(0) \tag{2.22}$$

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

$$\tau_{\exp,A} = \limsup_{t \to \infty} \frac{t}{-\log |\rho_{AA}(t)|} \tag{2.23}$$

and

$$\tau_{\exp} = \sup_{A} \tau_{\exp,A} \tag{2.24}$$

Thus, $\tau_{\exp}$ is the relaxation time of the slowest mode in the system. (If the state space is infinite, $\tau_{\exp}$ 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 $l^2(\pi)$.[11] 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:

$$R \equiv \inf\{r: \operatorname{spec}(P \restriction \mathbf{1}^{\perp}) \subset \{\lambda: |\lambda| \leqslant r\}\} \tag{2.25}$$

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

Facts (a)–(c) are a generalized Perron–Frobenius theorem[50]; fact (d) is a consequence of a generalized spectral radius formula.[51] Note that the rate of convergence to equilibrium from an initial nonequilibrium distribution is controlled by R, and hence by $\tau_{\exp}$.

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

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

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

$$= \frac{1}{2} + \sum_{t=1}^{\infty} \rho_{AA}(t) \tag{2.26}$$

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

$$\bar{A} \equiv \frac{1}{n} \sum_{t=1}^{n} A_t \tag{2.27}$$

has variance

$$\text{var}(\bar{A}) = \frac{1}{n^2} \sum_{r,s=1}^{n} C_{AA}(r-s) \tag{2.28}$$

$$= \frac{1}{n} \sum_{t=-(n-1)}^{n-1} \left(1 - \frac{|t|}{n}\right) C_{AA}(t) \tag{2.29}$$

$$\approx \frac{1}{n} (2\tau_{\text{int},A})\, C_{AA}(0) \qquad \text{for} \quad n \gg \tau \tag{2.30}$$

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

In summary, the autocorrelation times τ_{exp} and $\tau_{\text{int},A}$ 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, $n_{\text{disc}} \approx 20\tau_{\text{exp}}$ is usually more than adequate. On the other hand, $\tau_{\text{int},A}$ determines the statistical errors in Monte Carlo measurements of $\langle A \rangle$, once equilibrium has been attained.

Most commonly it is assumed that τ_{exp} and $\tau_{\text{int},A}$ are of the same order of magnitude, at least for "reasonable" observables A. But this is *not* true in general. In fact, one usually expects the autocorrelation function $\rho_{AA}(t)$ to obey a dynamic scaling law[52] of the form

$$\rho_{AA}(t;\beta) \sim |t|^{-a} F((\beta - \beta_c)\,|t|^b) \tag{2.31}$$

valid in the region

$$|t| \gg 1, \qquad |\beta - \beta_c| \ll 1, \qquad |\beta - \beta_c|\, |t|^b \quad \text{bounded} \tag{2.32}$$

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 in continuous and strictly positive, with $F(x)$ decaying rapidly (e.g. exponentially) as $|x| \to \infty$. Then it is not hard to see that

$$\tau_{\exp, A} \sim |\beta - \beta_c|^{-1/b} \tag{2.33}$$

$$\tau_{\text{int}, A} \sim |\beta - \beta_c|^{-(1-a)/b} \tag{2.34}$$

$$\rho_{AA}(t; \beta = \beta_c) \sim |t|^{-a} \tag{2.35}$$

so that $\tau_{\exp, A}$ and $\tau_{\text{int}, A}$ have *different* critical exponents unless $a = 0$.[12] Actually, this should not be surprising: replacing "time" by "space," we see that $\tau_{\exp, A}$ is the analogue of a correlation length, while $\tau_{\text{int}, A}$ is the analogue of a susceptibility; and (2.33)–(2.35) 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:

(B′) For each pair $x, y \in S$,

$$\pi_x p_{xy} = \pi_y p_{yx} \tag{2.36}$$

[Summing (B′) over x, we recover (B).] (B′) is called the *detailed-balance condition*; a Markov chain satisfying (B′) is called *reversible*.[13] (B′) is equivalent to the *self-adjointness* of P as on operator on the space $l^2(\pi)$. In this case, it follows from the spectral theorem that the autocorrelation function $C_{AA}(t)$ has a spectral representation

$$C_{AA}(t) = \int_{-1}^{1} \lambda^{|t|}\, d\sigma_{AA}(\lambda) \tag{2.37}$$

with a *nonnegative* spectral weight $d\sigma_{AA}(\lambda)$ supported on the interval $[-e^{-1/\tau_{\exp, A}}, e^{-1/\tau_{\exp, A}}]$. It follows that

$$\tau_{\text{int}, A} \leqslant \frac{1}{2}\left(\frac{1 + e^{-1/\tau_{\exp, A}}}{1 - e^{-1/\tau_{\exp, A}}}\right) \leqslant \frac{1}{2}\left(\frac{1 + e^{-1/\tau_{\exp}}}{1 - e^{-1/\tau_{\exp}}}\right) \approx \tau_{\exp} \tag{2.38}$$

[12] Our discussion of this topic in ref. 10 is incorrect.

[13] For the physical significance of this term, see Kemeny and Snell (ref. 53, Section 5.3) or Iosifescu (ref. 54, Section 4.5).

3. THE ALGORITHM

3.1. The Local (BFACF) Algorithm

The BFACF algorithm[14] is a Markov chain with state space $\mathscr{S}_N(x)$ and invariant probability distribution

$$\pi_\beta(\omega) = \Xi(\beta, x)^{-1} |\omega| \beta^{|\omega|} \tag{3.1}$$

where $|\omega|$ denotes the number of bonds in the walk ω, and

$$\Xi(\beta, x) = \sum_{N=0}^{\infty} N\beta^N c_N(x) \tag{3.2}$$

is the partition function for this ensemble. The elementary moves of the BFACF algorithm are the local deformations shown schematically in Fig. 1; they correspond to moving the middle bond by one lattice unit perpendicular to itself in one of the $2d-2$ possible directions. The moves (A), (B), and (C) change the number of bonds in the walk by $+2$, -2, and 0, respectively. One iteration of the BFACF algorithm consists of the following operations:

1. Choose at random a bond of the current walk ω (with equal probability for each bond).
2. Enumerate the $2d-2$ possible deformations of that bond; choose randomly among these deformations, giving each deformation a

[14] The description given in this section supersedes that of ref. 9, which suffers from an unfortunate confusion regarding the meaning of $p(\Delta N)$.

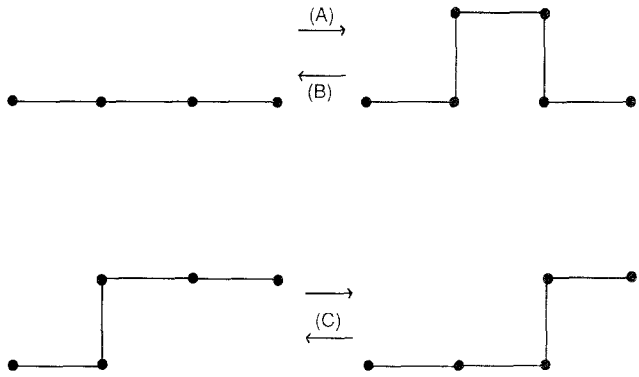


Fig. 1. The local (BFACF) moves: (A) $\Delta N = +2$; (B) $\Delta N = -2$; (C) $\Delta N = 0$.

probability $p(\Delta N)$ depending only on $\Delta N \equiv |\omega'| - |\omega|$. (If the sum of these probabilities is $s < 1$, then make a "null transition" $\omega \to \omega$ with probability $1 - s$.) The probabilities $p(\Delta N)$ will be specified below.

3. Check whether the proposed new walk ω' is self-avoiding. If it is, keep it; otherwise, make a null transition.

It follows that the transition matrix for the BFACF algorithm is given by

$$P(\omega \to \omega') = \frac{1}{|\omega|}\, p(|\omega'| - |\omega|)\, \chi_{\mathrm{SAW}}(\omega') \tag{3.3}$$

for $\omega' \neq \omega$, where

$$\chi_{\mathrm{SAW}}(\omega') = \begin{cases} 1 & \text{if } \omega' \text{ is self-avoiding} \\ 0 & \text{if } \omega' \text{ is not self-avoiding} \end{cases} \tag{3.4}$$

In order that this algorithm make sense, we must impose the inequalities

$$\sum_{\omega' \neq \omega} P(\omega \to \omega') \leqslant 1 \tag{3.5}$$

for all ω. To see what this implies, consider how step 2 of the algorithm is implemented. Once a bond has been chosen, we compare its direction with the directions of the preceding and following bonds along the walk. There are four possible cases (see Fig. 2):

(i) The directions of all three bonds are the same. In this case all $2d-2$ possible deformations have $\Delta N = +2$.

(ii) One neighbor bond has the same direction as the chosen bond, while the other is perpendicular. In this case one possible deformation has $\Delta N = 0$; the other $2d-3$ deformations have $\Delta N = +2$.

(iii) Both neighbor bonds are perpendicular to the chosen bond, and they are antiparallel to each other. In this case one possible deformation has $\Delta N = -2$; the other $2d-3$ deformations have $\Delta N = +2$.

(iv) Both neighbor bonds are perpendicular to the chosen bond, and they are either parallel or perpendicular to each other. In this case two possible deformations have $\Delta N = 0$; the other $2d-4$ deformations have $\Delta N = +2$.

[If the chosen bond is the first or last bond of the walk, then there is only one neighbor bond. If one pretends that the nonexistent ("phantom")

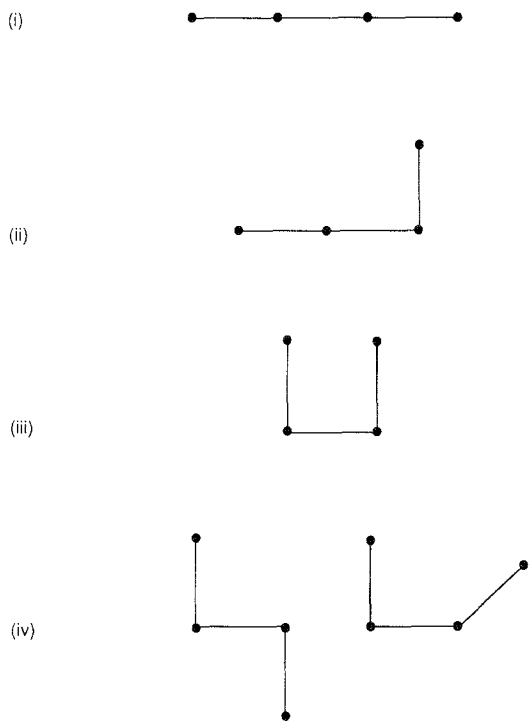


Fig. 2. The four possible cases of link orientations in a local (BFACF) move.

neighbor bond has the same direction as the chosen bond, then the above classification gives the correct answer.] We obtain, therefore, the following restrictions on $p(\Delta N)$:

$$(2d-2)\, p(+2) \leqslant 1 \qquad \text{case (i)} \tag{3.6}$$

$$p(0) + (2d-3)\, p(+2) \leqslant 1 \qquad \text{case (ii)} \tag{3.7}$$

$$p(-2) + (2d-3)\, p(+2) \leqslant 1 \qquad \text{case (iii)} \tag{3.8}$$

$$2p(0) + (2d-4)\, p(+2) \leqslant 1 \qquad \text{case (iv)} \tag{3.9}$$

However, inequality (ii) is a consequence of inequalities (i) and (iv) (just take the half-sum), so we can forget about (ii). Now, we must also satisfy the detailed-balance condition

$$\pi_\beta(\omega)\, P(\omega \to \omega') = \pi_\beta(\omega')\, P(\omega' \to \omega) \tag{3.10}$$

for the probability measure π_β defined in (3.1). This imposes the condition

$$p(+2) = \beta^2 p(-2) \tag{3.11}$$

Since $0 \leqslant \beta \leqslant \beta_c \equiv \mu^{-1} \leqslant d^{-1} < 1$, it follows that inequality (i) is strictly weaker than inequality (iii). Eliminating $p(-2)$ in favor of $p(+2)$, the two remaining inequalities are

$$[1 + (2d-3)\beta^2]\, p(+2) \leqslant \beta^2 \qquad \text{case (iii)} \tag{3.12}$$

$$2p(0) + (2d-4)\, p(+2) \leqslant 1 \qquad \text{case (iv)} \tag{3.13}$$

These inequalities determine a convex region in the $(p(0), p(+2))$ plane (see Fig. 3). Any point in this region defines a valid version of the BFACF algorithm; the choice between these versions should be made on the grounds of efficiency.

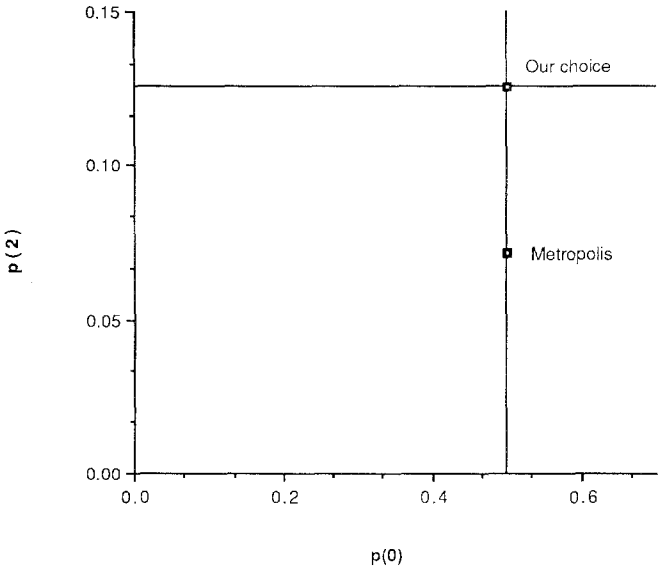
It makes sense physically that increasing the transition rates in a Monte Carlo algorithm (while keeping the same invariant measure π) can only improve the equilibration. Indeed, it is not hard to prove rigorously (see Theorems A.1–A.3 in the Appendix) that for *reversible* Markov chains, all autocorrelation times ($\tau_{\exp}$ and $\tau_{\text{int},A}$) are reduced (or at least stay constant) whenever the off-diagonal elements of the transition matrix P are increased (keeping the same π). It follows that $p(0)$ and $p(+2)$ should be made as large as possible. In the case $d=2$, this criterion determines a unique optimal point, namely the intersection of lines (iii) and (iv) (see Fig. 3a). In the case $d>2$, all we can say is that the optimal point(s) must lie somewhere on line (iv) below the intersection with line (iii) (see Fig. 3b). However, it is clearly bad to take $p(+2)$ *too* small, since this will slow down the transitions between walks of different length [indeed, for $p(+2)=0$ the algorithm becomes nonergodic]. It seems reasonable, therefore, to employ in all cases the algorithm defined by the intersection of lines (iii) and (iv), namely

$$p(-2) = \frac{1}{1 + (2d-3)\beta^2} \tag{3.14}$$

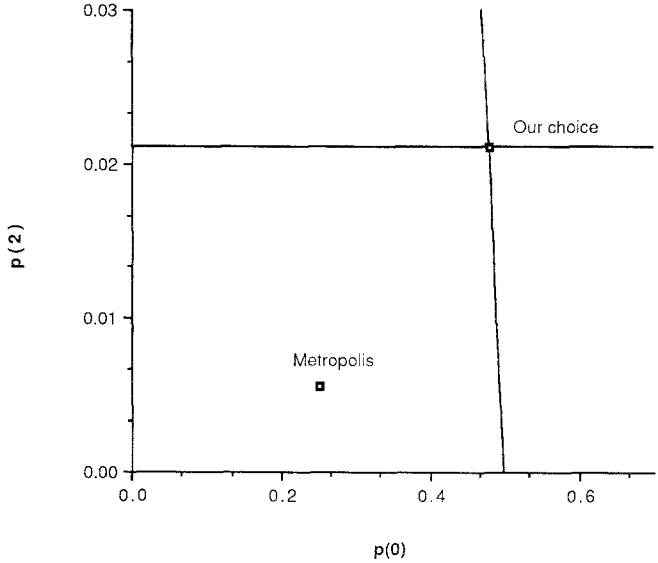
$$p(0) = \frac{1 + \beta^2}{2[1 + (2d-3)\beta^2]} \tag{3.15}$$

$$p(+2) = \frac{\beta^2}{1 + (2d-3)\beta^2} \tag{3.16}$$

Indeed, it can be shown (see Theorem A.3 and the example following it)



(a)



(b)

Fig. 3. The allowable region for $p(0)$ and $p(+2)$ in the BFACF algorithm. (a) Dimension $d=2$. (b) Dimension $d>2$. Region is drawn for $\beta=\beta_c$ for $d=2, 3$, respectively. "Our choice" denotes (3.14)–(3.16); "Metropolis" denotes (3.18).

that this choice is no more than a factor $[1+(2d-3)\beta^2]/(1+\beta^2)$ worse than the optimal choice; and

$$\frac{1+(2d-3)\beta^2}{1+\beta^2} \leqslant \frac{1+(2d-3)\beta_c^2}{1+\beta_c^2} = \begin{cases} \approx 1.087 & d=3 \\ \approx 1.085 & d=4 \\ \leqslant 1+(2d-4)/(d^2+1) \leqslant 1.2 & \text{any } d \end{cases} \tag{3.17}$$

where we have used $\beta_c \approx 0.2135$ $(d=3)$,[55] $\beta_c \approx 0.1477$ $(d=4)$,[56] and $\beta_c \leqslant 1/d$ (all d). We have therefore adopted (3.14)–(3.16) in refs. 10 and 11 and in the present paper.

Remarks. 1. A perhaps simpler version of the BFACF algorithm is to choose at random a link k *and* a deformation direction $e_\perp$; the proposed deformation is then accepted or rejected according to the Metropolis criterion, i.e., with acceptance probability

$$\begin{aligned} a(\omega \to \omega') &= \min\left[\frac{\pi(\omega')}{\pi(\omega)}, 1\right] \\ &= \chi_{\mathrm{SAW}}(\omega') \times \begin{cases} 1 & \text{if } \Delta N = 0 \text{ or } -2 \\ \beta^2 & \text{if } \Delta N = +2 \end{cases} \end{aligned} \tag{3.18}$$

This amounts to choosing $p(0)=p(-2)=1/(2d-2)$, $p(+2)=\beta^2/(2d-2)$, which is always inferior to the choice (3.14)–(3.16).

2. Our decision in the case $d>2$ to maximize $p(+2)$ rather than $p(0)$ is also sensible in the context of our "hybrid" algorithm: the nonlocal moves hopefully assure the equilibration of walks at fixed N, so the $\Delta N=0$ local moves are somewhat redundant; while the $\Delta N=+2$ moves are crucial in assuring equilibration between walks of different lengths.

It is also necessary, of course, to verify that the BFACF algorithm is ergodic, i.e., that it is possible to get from any element of $\mathscr{S}_N(x)$ to any other by some finite sequence of allowed local deformations. The situation is at present rather complicated:

(a) In dimension $d=2$, Madras[57] has proven that the BFACF algorithm is ergodic, for all choices of x.

(b) In dimension $d=3$, the algorithm is *nonergodic* due to the possibility of knots (a conserved topological quantity) if

$$\|x\|_\infty \equiv \max(|x_1|, |x_2|, |x_3|) = 1 \tag{3.19}$$

If $\|x\|_\infty \geqslant 2$, we do not know whether the algorithm is ergodic: in order to prove it, one would have to show that any "knot," no

matter how large and complicated, can be "disentangled" by a motion which never passes more than one strand at a time between the endpoints.

(c) In dimension $d \geq 4$, it is not known whether the BFACF algorithm is ergodic: we suspect that it is, but to prove it will require a capacity for multidimensional visualization that exceeds our own.

The dynamical behavior of the BFACF algorithm is rather peculiar (and not completely understood at present). A plausible heuristic argument[10] suggests that $\tau \sim \langle N \rangle^{2+2\nu}$, but this seems to be *false*![15] Indeed, Sokal and Thomas[22] have proven the surprising result that $\tau_{\exp} = +\infty$ for *all* $\beta > 0$. The proof is both simple and physically illuminating, so we reproduce it here.

The result is actually a corollary of a more general theorem about Markov chains. 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.,

$$T_{AB} \equiv \min\{n: p_{xy}^{(n)} > 0 \text{ for some } x \in A,\ y \in B\} \tag{3.20}$$

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 $\tau_{\exp}$ must be large. More precisely:

Theorem 1. Consider a Markov chain with transition matrix P satisfying detailed balance for the probability measure π. Let T_{AB} be defined as in (3.20). Then

$$\tau_{\exp} \geq \sup_{A,B \in S} \frac{2(T_{AB} - 1)}{-\log[\pi(A)\,\pi(B)]} \tag{3.21}$$

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

$$(\chi_A, P^n \chi_B)_{l^2(\pi)} \equiv \sum_{\substack{x \in A \\ y \in B}} \pi_x p_{xy}^{(n)} = 0 \tag{3.22}$$

[15] We are less convinced now than we were 3 years ago about the plausibility of this heuristic argument. For the *fixed-N* local-deformation algorithms, the lower bound $\tau \gtrsim N^{2+2\nu}$ is clearly valid (see Example 3 following Theorem A.7), and it is reasonable to conjecture that it is close to sharp. However, for the BFACF algorithm the argument is far less plausible, since the $\Delta N = \pm 2$ moves could in principle cause the center-of-mass vector to relax in a time as small as $\langle N \rangle^2$—much smaller than $\langle N \rangle^{2+2\nu}$.

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

$$(\chi_A - \pi(A)\mathbf{1},\, P^n[\chi_B - \pi(B)\mathbf{1}])_{l^2(\pi)} = -\pi(A)\,\pi(B) \tag{3.23}$$

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

$$\|P^n \restriction \mathbf{1}^\perp\| = \|P \restriction \mathbf{1}^\perp\|^n = R^n \tag{3.24}$$

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

$$\begin{aligned}
&|(\chi_A - \pi(A)\mathbf{1},\, P^n[\chi_B - \pi(B)\mathbf{1}])_{l^2(\pi)}| \\
&\quad \leqslant R^n \|\chi_A - \pi(A)\mathbf{1}\|_{l^2(\pi)}\, \|\chi_B - \pi(B)\mathbf{1}\|_{l^2(\pi)} \\
&\quad = R^n \pi(A)^{1/2}[1 - \pi(A)]^{1/2}\, \pi(B)^{1/2}[1 - \pi(B)]^{1/2} \\
&\quad \leqslant R^n \pi(A)^{1/2}\, \pi(B)^{1/2}
\end{aligned} \tag{3.25}$$

Combining (3.23) with (3.25) and taking $n = T_{AB} - 1$, we arrive after a little algebra at (3.21). ∎

To apply this theorem to the BFACF algorithm, let ω^* be a fixed short walk from 0 to x, and let ω^n be a quasirectangular walk from 0 to x of linear size $\approx n$ (Fig. 4). 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

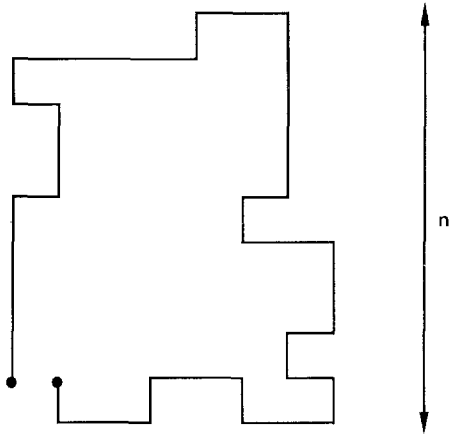


Fig. 4. A large quasirectangular walk of linear size $\approx n$.

$\omega^n \cup \omega^*$ can change by at most one unit in a local deformation. Applying the theorem with $A = \{\omega^n\}$ and $B = \{\omega^*\}$, we obtain

$$\tau_{\exp} \geqslant \sup_n \frac{\sim n^2}{\sim n} = +\infty \tag{3.26}$$

The BFACF algorithm is characterized, therefore, by arbitrarily slowly-relaxing modes associated with transitions $\omega \to \omega'$ that have $\mathscr{A}(\omega, \omega') \gg \max(|\omega|, |\omega'|)$, where $\mathscr{A}(\omega, \omega')$ is the minimum surface area spanned by the union of ω and ω'. Consequently, one expects that for "most" observables A, the autocorrelation function $\rho_{AA}(t)$ will decay *non-exponentially* as $t \to \infty$, so that $\tau_{\exp, A} = \infty$. However, there is nothing to prevent $\tau_{\text{int}, A}$ from being finite, and indeed one expects that $\tau_{\text{int}, A} < \infty$ for "reasonable" observables A, i.e., those that are not too strongly coupled to very long walks. It then makes sense to study the dynamic critical exponent p_A defined by

$$\tau_{\text{int}, A} \sim \langle N \rangle^{p_A} \tag{3.27}$$

for $\beta \uparrow \beta_c$. A preliminary study[10] found $p_A = 3.0 \pm 0.4$ for $A = N, N^2, N^3$ in the two-dimensional SAW, and $p_A = 2.2 \pm 0.5$ for the same observables in the two-dimensional nonreversal random walk (NRRW). We present additional data on the SAW in Section 4. Also, in the Appendix we prove the rigorous lower bound $\tau_{\text{int}, \mathscr{A}} \geqslant \text{const} \times \langle \mathscr{A}^2 \rangle$ and a similar bound for $\tau_{\text{int}, |\mathscr{A}|}$ (see Example 1 following Theorem A.7). Assuming the usual scaling behavior with exponent given by (2.13), this implies

$$p_{\mathscr{A}}, p_{|\mathscr{A}|} \geqslant 4\nu = \begin{cases} 3 & \text{for the } d = 2 \text{ SAW} \\ \approx 2.4 & \text{for the } d = 3 \text{ SAW} \\ 2 & \text{for the } d \geqslant 4 \text{ SAW} \\ 2 & \text{for the NRRW (any } d) \end{cases} \tag{3.28}$$

In the absence of any additional physical mechanisms for "slow modes," it is reasonable to expect that this bound is close to sharp, and this is confirmed by our numerical estimates for the $d = 2$ SAW and NRRW.

3.2. The Nonlocal Algorithm

As we have just seen, the BFACF algorithm has very slowly-relaxing modes associated with transitions to very long walks whose "surface area" is much greater than their length. In this subsection we describe how to supplement the BFACF algorithm with *nonlocal* moves that are specifically designed to speed up these slow modes. These nonlocal moves are

N-conserving; ideally, they would cause instant equilibration among the walks at fixed N, leaving to the local moves the task of bringing about equilibration between the spaces of different N. Therefore, in evaluating the performance of the hybrid (local + nonlocal) algorithm, two distinct questions arise:

(a) How well does the *idealized* hybrid algorithm perform? That is: *if* the nonlocal moves were to bring about instant equilibration among the walks at fixed N, what would be the dynamic critical behavior of the hybrid algorithm?

(b) How well does a particular set of nonlocal moves approximate the ideal of instant equilibrium? To what extent does the non-ideality of the nonlocal moves degrade the performance of the combined algorithm?

Our numerical experiments (Section 4) are designed to disentangle the answers to these two questions.

The particular nonlocal moves that we shall consider here are "cut-and-paste" moves which cut the walk into two or more pieces, permute and/or invert the pieces, and then reassemble them. In order to describe these moves, it is convenient to think of a walk $\omega = (\omega_0, \omega_1, \omega_2, ..., \omega_N)$ as a sequence of steps $\mathbf{a}_i(\omega) \equiv \omega_i - \omega_{i-1}$ $(1 \leqslant i \leqslant N)$. We then define the following operations:

1. *Subwalk.* If $\omega = (\omega_0, \omega_1, \omega_2, ..., \omega_N)$, then let $\omega^{i,j}$ be the part of ω from point i through point j $(0 \leqslant i \leqslant j \leqslant N)$, i.e.,

$$\omega^{i,j} = (\omega_i, \omega_{i+1}, ..., \omega_{j-1}, \omega_j) \tag{3.29}$$

2. *Concatenation.* If $\omega = (\omega_0, \omega_1, ..., \omega_N)$ and $\omega' = (\omega'_0, \omega'_1, ..., \omega'_{N'})$, then let $\omega \circ \omega'$ be the walk obtained by concatenating ω and ω', i.e.,

$$(\omega \circ \omega')_i = \begin{cases} \omega_i - \omega_0 & \text{if } 0 \leqslant i \leqslant N \\ (\omega_N - \omega_0) + (\omega'_{i-N} - \omega'_0) & \text{if } N \leqslant i \leqslant N + N' \end{cases} \tag{3.30}$$

Clearly, the steps of $\omega \circ \omega'$ are given by

$$\mathbf{a}_i(\omega \circ \omega') = \begin{cases} \mathbf{a}_i(\omega) & \text{if } 1 \leqslant i \leqslant N \\ \mathbf{a}_{i-N}(\omega') & \text{if } N+1 \leqslant i \leqslant N+N' \end{cases} \tag{3.31}$$

3. *Inversion.* If $\omega = (\omega_0, \omega_1, \omega_2, ..., \omega_N)$, let $I\omega$ be the walk defined by

$$(I\omega)_i = \omega_N - \omega_{N-i} \tag{3.32}$$

Equivalently, $I\omega$ is obtained by inverting the order of steps in ω, i.e.,

$$\mathbf{a}_i(I\omega) = \mathbf{a}_{N-i+1}(\omega) \tag{3.33}$$

In particular, $I\omega$ has the same end-to-end distance vector as ω: $\omega_N - \omega_0 = (I\omega)_N - (I\omega)_0$.

4. *Permutation.* Let $\omega = (\omega_0, \omega_1, \omega_2, ..., \omega_N)$ and $0 \leqslant i \leqslant N$. Then $P_i\omega$ is the walk obtained by cutting ω at site i and permuting the two pieces:

$$P_i\omega = \omega^{i,N} \circ \omega^{0,i} \tag{3.34}$$

The steps of $P_i\omega$ are clearly those of ω taken in the order $\mathbf{a}_{i+1}$, $\mathbf{a}_{i+2}, ..., \mathbf{a}_N, \mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_i$.

We have studied two slightly different nonlocal algorithms: one using a single pivot point, and one using a pair of pivot points. Let us consider first the 1-pivot algorithm. We first choose randomly a location i along the walk ($0 \leqslant i \leqslant N$) to serve as the pivot point. We now consider splitting the walk at i into its two subwalks $\omega^{0,i}$ and $\omega^{i,N}$, inverting and/or permuting the two pieces, and then reassembling the walk. Clearly, there are $8 = 2^3$ possible outcomes, according as the first (resp. second) subwalk is or is not inverted, and the two subwalks are or are not permuted. However, these eight outcomes fall into four equivalence classes, with the two walks in each class being related by an overall inversion (a trivial symmetry operation). Therefore, it suffices to choose one representative from each of the four equivalence classes. In order to minimize the computer time, we choose in all cases to invert the *shorter* of the two subwalks. The nonidentity operations are therefore reduced to three (see Fig. 5):

1. *Permute*: $\omega \to \omega' \equiv P_i\omega$.
2. *Invert*:

$$\omega \to \omega' \equiv I_i\omega \equiv \begin{cases} \omega^{0,i} \circ I\omega^{i,N} & \text{if } i \geqslant N/2 \\ I\omega^{0,i} \circ \omega^{i,N} & \text{otherwise} \end{cases} \tag{3.35}$$

3. *Invert and permute*:

$$\omega \to \omega' \equiv P_i I_i \omega \equiv \begin{cases} I\omega^{i,N} \circ \omega^{0,i} & \text{if } i \geqslant N/2 \\ \omega^{i,N} \circ I\omega^{0,i} & \text{otherwise} \end{cases} \tag{3.36}$$

We choose randomly (with equal probability) one of these three operations, and compute whether the proposed new walk ω' is indeed self-avoiding. (See Section 3.3 for details of how this computation is

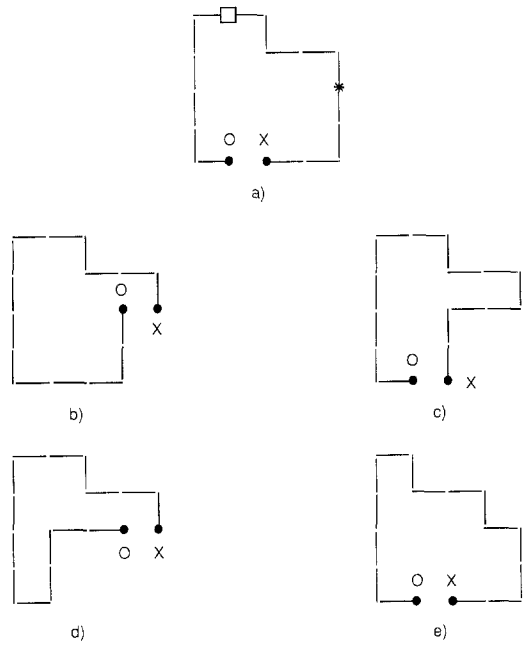


Fig. 5. The cut-and-paste moves applied to the walk shown in (a). The star denotes the pivot for the following moves: (b) a permutation; (c) an inversion; (d) a combined inversion/permutation. (e) The 2-pivot inversion, using the site denoted in (a) by an open square as the second pivot.

performed.) If ω' is self-avoiding, we accept it; otherwise, we make a null transition. It is clear that the 1-pivot move is N-conserving and satisfies detailed balance with respect to any probability measure giving equal weight to each N-step SAW [in particular, the distribution (3.1)].

We remark that two successive permutations with different pivot sites are equivalent to a single permutation:

$$P_j P_i \omega = P_k \omega \qquad \text{with} \quad k = i + j \,(\text{mod } N) \tag{3.37}$$

As mentioned previously, the motivation for the cut-and-paste moves is to speed up the slow modes of the BFACF algorithm. Indeed, it easy to see that an inversion (or an inversion/permutation) can produce a change of order N^2 in the area spanned by a large rectangular configuration, producing a walk that can be reduced to a short walk by $\sim N$ BFACF moves (rather than $\sim N^2$ of them).

On the other hand, a permutation can cause the area $\mathcal{A}(\omega)$ spanned by a walk to change by at most an amount of order N. To see this,

consider first the case of a self-avoiding *polygon*, i.e., a *closed* loop. Then a permutation is simply a redefinition of the origin, which does not change the area at all. Consider next the case of a SAW with endpoints which are nearest neighbors. The area of this object is obtained by appending an additional step from the final point back to the initial point, thereby creating a closed loop. A permutation is equivalent to moving this "phantom" step to a different location with the loop, so that the area is changed by that of the strip swept out by this motion, which is at most equal to the diameter of the loop, i.e., of order N. A similar argument shows that for SAWs with endpoints separated by a distance x, the area change is at most Nx. It follows that the permutations alone are not terribly efficient in destabilizing large quasirectangular configurations; it is necessary to use inversions as well.

Let us now consider the 2-pivot algorithm. We begin by choosing randomly a pair of locations i, j along the walk ($0 \leqslant i < j \leqslant N$) to serve as the pivot points. Define now the operation $I_{i,j}$ which inverts the middle segment of the walk:

$$I_{i,j}\omega \equiv \omega^{0,i} \circ I\omega^{i,j} \circ \omega_{j,N} \tag{3.38}$$

The proposed new walk $\omega' \equiv I_{i,j}\omega$ is then tested for self-avoidance: if it is self-avoiding, we accept it; otherwise, we make a null transition. Clearly this 2-pivot move is N-conserving and satisfies detailed balance.

Let us now examine the relation between the 1-pivot and 2-pivot moves. On the one hand, the 1-pivot inversion is clearly a special case of the 2-pivot inversion:

$$I_i \equiv \begin{cases} I_{i,N} & \text{if} \quad i \geqslant N/2 \\ I_{0,i} & \text{otherwise} \end{cases} \tag{3.39}$$

(The 1-pivot permutation clearly cannot be expressed in terms of 2-pivot inversions.) On the other hand, the 2-pivot inversion is equivalent (modulo an overall inversion) to a composition of 1-pivot moves:

$$I_{i,j} \equiv \begin{cases} (P_{N-i}I_{j-i})P_i & \text{if} \quad j-i < N/2 \\ I(P_iI_{j-i})P_i & \text{otherwise} \end{cases} \tag{3.40}$$

Notice, however, that the intermedciate walk in the sequence of two 1-pivot moves might not be self-avoiding. For example, if both ends of the walk are in *culs-de-sac*, then no permutation is possible, so that the correspondence (3.40) cannot be realized. In fact, for the configuration shown in Fig. 6, *no* 1-pivot move is possible, although 2-pivot moves can work perfectly well.[16] One might suspect, however, that such configurations are

[16] We thank Neal Madras for pointing out this possibility.

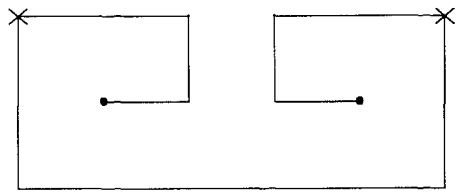


Fig. 6. A "double cul-de-sac" configuration for which 1-pivot moves are impossible. A choice for the pivots of a possible 2-pivot move is indicated by crosses.

relatively rare. If so, then the 2-pivot algorithm cannot be *much* better than the 1-pivot algorithm (because any 2-pivot move would occur anyway with reasonable probability in the 1-pivot algorithm), although it can be *somewhat* better because of the greater randomness in the choice of pivot points. Our numerical data (Section 4) bear out this expectation.

Let us mention that both the 1-pivot and 2-pivot moves conserve not only the total number of links (N), but also the numbers of links in each direction ($N_1^{\pm}, N_2^{\pm}, \ldots, N_d^{\pm}$). Therefore, these pivot moves do *not* define an algorithm that is ergodic on each space $\mathscr{S}_N(x)$, so they cannot fulfill our hope of bringing about instant equilibration among the walks at fixed N. However, they may still be capable of bringing about a good enough equilibration so that the remaining slow modes (those involving changes in $N_1^{\pm}, N_2^{\pm}, \ldots, N_d^{\pm}$) can be brought about efficiently by the local (BFACF) moves. We discuss this further, in light of our numerical data, in Section 4. Let us also mention that Dubins *et al.*,[28] Madras *et al.*,[29] and Janse van Rensburg *et al.*[30] have devised *ergodic* algorithms for fixed-N, fixed-endpoint SAWs (in any dimension), using the 2-pivot inversion (3.38) together with other 2-pivot moves involving diagonal reflections and axis interchanges.

Finally, Madras *et al.*[29] have proven that the hybrid BFACF/2-pivot algorithm is ergodic, in any dimension and for any $x \neq 0$.

3.3. Data Structures and Computational Complexity

In this section we discuss the data structures that we use to represent the walks, and we analyze the computational complexity of the BFACF and cut-and-paste moves. The principal goal is to devise a data structure in which *both* types of moves can be implemented efficiently—that is, the BFACF moves in a time of order 1, and the cut-and-paste moves in a time of order N or less.

Let us look first at the BFACF moves. The three principal operations are:

1. Select at random a link.
2. Insert one or two sites (for a $\Delta N = 0$ or $+2$ move, respectively).
3. Delete one or two sites (for a $\Delta N = 0$ or -2 move, respectively).

Suppose first that the walk $\omega = (\omega_0, \omega_1, ..., \omega_N)$ were stored as a sequentially allocated linear list $\{s(i)\}_{i=0}^{N}$, where $s(i)$ is an integer that codes the coordinates of the site ω_i. Then selection of a random link would be easy (time of order 1), but insertion and deletion of links would be very costly (time of order N) because of the need for "garbage collection" to keep the list ordered. Suppose, alternatively, that the walk ω were stored as a linked (or doubly linked) linear list dispersed inside some large block of memory. Then insertion and deletion would be easy (time of order 1), but selection of a random link would be very costly (time of order N) because of the need to "thread through" the list sequentially.

To get the best of both worlds, we use a *contiguously allocated, doubly linked linear list*. That is, the walk coordinates are stored in a contiguous array $\{s(i)\}_{i=0}^{N}$, but *not in any particular order*; that is, $s(i)$ codes *some* site of the walk, but *not* necessarily ω_i. To keep track of the sequence of steps along the walk, we use forward pointers $\{p^+(i)\}_{i=0}^{N}$ and backward pointers $\{p^-(i)\}_{i=0}^{N}$; here $p^+(i)$ [resp. $p^-(i)$] is the index corresponding to the site following (resp. preceding) the site whose index is i, or -1 if no such site exists. The initial and final points of the walk, which are fixed, are by convention allocated to indices 0 and 1, respectively. Therefore,

$$\begin{aligned} s(0) &\text{ codes } \omega_0 \,(=0) \\ s(p^+(0)) &\text{ codes } \omega_1 \\ s(p^+(p^+(0)) &\text{ codes } \omega_2 \\ &\vdots \end{aligned}$$

and so on. Likewise,

$$\begin{aligned} s(1) &\text{ codes } \omega_N \,(=x) \\ s(p^-(1)) &\text{ codes } \omega_{N-1} \\ s(p^-(p^-(1))) &\text{ codes } \omega_{N-2} \\ &\vdots \end{aligned}$$

and so on. (See Table I.) We also keep a list of integers $\{b(i)\}_{i=0}^{N-1}$ with values in $\{1, 2, ..., 2d\}$, which code the direction of the step of the walk following site $s(i)$; we enumerate the directions in such a way that direction $2d+1-l$ is opposite to direction l. In principle, the list $\{b(i)\}_{i=0}^{N-1}$ is redundant, since it can be computed by comparing $s(i)$ to $s(p^+(i))$, but its presence speeds up the program considerably.

Table I. A Typical Configuration of the Data Structure, Shown for a Walk Having $N=5$

i	$s(i)$	$p^+(i)$	$p^-(i)$
0	ω_0	5	-1
1	ω_5	-1	3
2	ω_2	4	5
3	ω_4	1	4
4	ω_3	3	2
5	ω_1	2	0

Finally, for purposes of self-avoidance checking we maintain a "bit table" in which each site of a large (512×512) square box is assigned one bit: this bit is set to 1 if the site is occupied by the walk, and 0 otherwise. Clearly such a table can be checked and updated in a time of order 1. [In dimension $d \geqslant 3$, such a bit table might require a prohibitively large memory; if so, a "hash table"[58,59] could be used instead (ref. 23, Section 3.4).] In principle, this means that we are simulating a SAW restricted to a 512×512 periodic box. However, in our simulations it is a very rare event for a walk to "reach around the box and touch itself," so the finite-size systematic errors are negligible.

Let us now examine in detail how a BFACF upgrading is performed. First we pick (with uniform probability) an index $i \in \{1, 2, ..., N\}$; then $(s(p^-(i)), s(i))$ is a random link from the walk ω. (Recall that the starting point of the walk is always assigned to index $i=0$.) Using the pointer arrays $p^{\pm}$ and the direction array b, we can classify the chosen link as case (i), (ii), (iii), or (iv) (see Section 3.1), and choose randomly (with the appropriate probabilities) a proposed deformation. There are then three possibilities:

(a) $\Delta N = +2$ *proposal.* The two proposed new sites are checked in the bit table. If both of them are currently vacant, then they are inserted into the linear list at indices $N+1$ and $N+2$, with the appropriate changes made to the arrays $p^{\pm}$ and b. The bit table is also updated.

(b) $\Delta N = 0$ *proposal.* The proposed new site is checked in the bit table. If it is vacant, then it is inserted into the linear list at the index currently occupied by the site that is to be removed, with appropriate changes to the array b. The bit table is also updated.

(c) $\Delta N = -2$ *proposal.* Two sites must be removed from the linear list; to keep the list contiguous, the entries that are currently in

the locations $N-1$ and N take their place. The arrays $p^{\pm}$ and b are updated appropriately. The bit table is also updated.

Clearly all these operations can be performed in a time of order 1 (i.e., independent of N). We remark that to speed up the definitions of the coordinates of the new points, we use a table which provides the (coded) coordinates for the neighbor to a given site in a given direction.

Important Note. The algorithm that we have actually implemented in the work reported in this paper is slightly different: we pick (with uniform probability) an index $i \in \{0, 1, ..., N\}$. If $s(i)$ is not the last point of the walk, then $(s(i), s(p^+(i)))$ is a random link from ω, and we proceed as before; otherwise we make a null transition. Because a given link is here chosen with probability $1/(N+1)$, it follows that we are implementing a variant of the BFACF algorithm in which the invariant probability distribution is

$$\pi_\beta(\omega) = \Xi(\beta, x)^{-1}(|\omega| + 1)\beta^{|\omega|} \tag{3.1a}$$

and the partition function is

$$\Xi(\beta, x) = \sum_{N=0}^{\infty} (N+1)\,\beta^N c_N(x) \tag{3.2a}$$

[instead of (3.1)–(3.2)]. This variant is slightly less efficient than the standard BFACF algorithm, because of null transitions occurring with probability $1/(N+1)$.

When nonlocal moves are performed, drastic changes occur in the lists s and b and in the bit table. The changes in the bit table are *tentative*: we do not know until the end whether the proposed new walk will be accepted. We have therefore found it convenient to maintain *two* bit tables: at any given time, one is *active* and the other is *scratch*; a flag indicates which is which. The self-avoidance checking is carried out by writing into the scratch bit table (which is initially empty). If the proposed new walk is accepted, the scratch bit table becomes the active one (i.e., the flag is flipped); otherwise, the flag remains unchanged. At the end, the bit table that is now scratch (whichever one that is) is cleared. In order to facilitate this clearing, we maintain at all times two linear lists that specify which words of the two bit tables have nonzero entries. Finally, it is convenient to maintain active and scratch arrays also for the lists s and b: the tentative new entries are computed during the process of self-avoidance checking, and are placed in the scratch arrays; these become the active arrays if the proposed walk is accepted. However, this active/scratch procedure is not necessary for the lists p^+ and p^-, since the changes to these lists in a successful nonlocal move are much less drastic: in a permutation only the

entries corresponding to the pivot and the endpoints of the walk need to be changed, while in an inversion no change in the lists $p^{\pm}$ need be made at all.

We now come to a very important fact: during the pasting of a walk cut into pieces, a self-intersection is most likely to occur *near* the site(s) where the pasting occurs, if one occurs at all. We therefore begin the construction of the new walk *at the pivot site(s)*, and continue by defining the other points alternating backward and forward along the walk (ref. 23, Section 3.4). As will be seen, this has an effect on the computational complexity: the mean CPU time for a *failed* nonlocal move can be arranged to grow only as a fractional power of N. (Of course, the CPU time for a *successful* nonlocal move is always of order N).

Let us now define precisely the algorithm that we have implemented. First, we choose a random index $i \in \{0,..., N\}$. Next, we choose randomly to make either a nonlocal move (with probability p_{nl}) or a BFACF move (with probability $1 - p_{nl}$). Then:

(a) *BFACF move.* If $s(i)$ is the last point of the walk (i.e., $i = 1$), we make an immediate rejection. Otherwise we carry out the BFACF move, as described above, on the link $(s(i), s(p^+(i)))$.

(b) *Nonlocal move.* This is different for the 1-pivot and 2-pivot algorithms:

(i) *1-pivot algorithm.* We choose randomly (with equal probability) to perform either an inversion, a permutation, or a combined inversion/permutation. In the latter two cases, an immediate rejection is made if $s(i)$ is either the first or last point of the walk ($i = 0$ or 1). In the case of an inversion, an immediate rejection is made if $s(i)$ is either the first, second, next-to-last, or last point of the walk [$i = 0$, $p^+(0)$, $p^-(1)$, or 1], since these moves are simply the identity. If an immediate rejection is not made, then we "thread through" the linear list, starting at index 0, in order to determine whether the chosen pivot site $s(i)$ is in the first or second half of the walk; this is needed, if an inversion is to be made, for determining which half of the walk should be inverted [cf. (3.35)–(3.36)].

(ii) *2-pivot algorithm.* We choose randomly a second index $j \in \{0,..., N\}$. An immediate rejection is made if $i = j$, since such a move is simply the identity. An immediate rejection is also made if $s(i)$ or $s(j)$ (or both) is either the first or last point of the walk (i or $j = 0$ or 1), since such a move is either an overall inversion or else is equivalent to a 1-pivot

move.[17] Finally, an immediate rejection is made if $s(i)$ and $s(j)$ are successive sites along the walk, since in that case the move is also the identity. If an immediate rejection is not made, then we "thread through" the linear list, starting at index 0, in order to determine which of the two sites $s(i)$, $s(j)$ comes first in the walk.

It is useful to define an *effective* probability for nonlocal moves $p_{nl,\mathrm{eff}}$ by disregarding those proposed nonlocal moves that suffer immediate rejections. A straightforward computation yields

$$1-\frac{p_{nl,\mathrm{eff}}}{p_{nl}}=\begin{cases}\left\langle\dfrac{8}{3(N+1)}\right\rangle-\dfrac{p_1}{3} & \text{for the 1-pivot algorithm}\\ \left\langle\dfrac{7N-5}{(N+1)^2}\right\rangle+\dfrac{p_1}{2} & \text{for the 2-pivot algorithm}\end{cases} \tag{3.41}$$

where $p_1=c_1(x)/\Xi(\beta, x)$ is the probability of a 1-step walk. Clearly the first terms in (3.41) behave like $\langle 1/N\rangle$. Note that, from (2.4),

$$\left\langle\frac{1}{N}\right\rangle\equiv\frac{\sum_{N=0}^{\infty}\beta^N c_N(x)}{\sum_{N=0}^{\infty}N\beta^N c_N(x)}\sim\begin{cases}\text{const}>0 & \text{if } \alpha_{\mathrm{sing}}<0\\ (\beta_c-\beta)^{\alpha_{\mathrm{sing}}} & \text{if } 0<\alpha_{\mathrm{sing}}<1\\ (\beta_c-\beta) & \text{if } \alpha_{\mathrm{sing}}>1\end{cases} \tag{3.42}$$

as $\beta\uparrow\beta_c$, while

$$\langle N\rangle\equiv\frac{\sum_{N=0}^{\infty}N^2\beta^N c_N(x)}{\sum_{N=0}^{\infty}N\beta^N c_N(x)}\sim\begin{cases}\text{finite} & \text{if } \alpha_{\mathrm{sing}}<-1\\ (\beta_c-\beta)^{-1-\alpha_{\mathrm{sing}}} & \text{if } -1<\alpha_{\mathrm{sing}}<0\\ (\beta_c-\beta)^{-1} & \text{if } \alpha_{\mathrm{sing}}>0\end{cases} \tag{3.43}$$

Therefore,

$$\left\langle\frac{1}{N}\right\rangle\sim\begin{cases}\text{const}>0 & \text{if } \alpha_{\mathrm{sing}}<0\\ \langle N\rangle^{-\alpha_{\mathrm{sing}}} & \text{if } 0<\alpha_{\mathrm{sing}}<1\\ \langle N\rangle^{-1} & \text{if } \alpha_{\mathrm{sing}}>1\end{cases} \tag{3.44}$$

Likewise,

$$p_1\sim(\beta_c-\beta)^{\alpha_{\mathrm{sing}}}\sim\begin{cases}\text{const}>0 & \text{if } \alpha_{\mathrm{sing}}<0\\ \langle N\rangle^{-\alpha_{\mathrm{sing}}} & \text{if } \alpha_{\mathrm{sing}}>0\end{cases} \tag{3.45}$$

[17] Our choice to reject when *one but not both* of $s(i)$ and $s(j)$ is an endpoint of the walk was motivated by the desire to study a "pure" 2-pivot algorithm, without 1-pivot moves. However, it would also be reasonable to allow such moves.

Now, in any dimension $d<4$ we expect that $\alpha_{\text{sing}}>0$, so $p_{nl,\text{eff}} \to p_{nl}$ as $\langle N \rangle \to \infty$. In Table II we report the measured values of $p_{nl,\text{eff}}$ as a function of p_{nl} and β, for the 1-pivot and 2-pivot algorithms in dimension $d=2$ (with endpoint $|x|=1$).

Let us now analyze the computational complexity of the nonlocal moves. Clearly a successful nonlocal move takes a time of order N, since it is neccessary to make $N+1$ insertions into the bit table in order to verify that the proposed new walk is self-avoiding. However, the failed nonlocal moves could well take an average time considerably less than N if self-intersections tend to be detected early (i.e., by checking $\ll N$ steps)—this is the motivation for constructing the proposed new walks starting at the pivot point(s) and working outward. In the pivot algorithm for *free*-endpoint SAWs, it was argued (ref. 23, Section 3.4) that the mean CPU time per failure behaves as $\sim N^{1-q}$, where q is the critical exponent for the acceptance fraction ($f \sim N^{-q}$). It is natural to expect a similar behavior for the cut-and-paste algorithm.

In our algorithm as currently implemented, however, all nonlocal moves have a contribution to CPU time that is proportional to N (albeit with a very small proportionality constant), because of the preliminary operations that involve "threading through" the linear list: for the 1-pivot algorithm, to determine whether the pivot site $s(i)$ is in the first or second half of the walk; and for the 2-pivot algorithm, to determine which of the two sites $s(i)$, $s(j)$ comes first in the walk. Asymptotically for large $\langle N \rangle$ it would be preferable to avoid these contributions: for the 1-pivot algorithm, by abandoning the insistence on inverting always the shorter segment of the walk; and for the 2-pivot algorithm, by arranging the computation so that it is not necessary to know *a priori* whether $s(i)$ precedes or follows

Table II. $p_{nl,\text{eff}}/p_{nl}$ at Various β for the One-Pivot and Two-Pivot Algorithms

		$p_{nl,\text{eff}}/p_{nl}$	
β	$\langle N \rangle$	1-Pivot	2-Pivot
0.3690	20.3	0.60	0.43
0.3728	33.2	0.67	0.52
0.3744	43.9	0.70	0.57
0.3760	65.4	0.74	0.63
0.3771	102	0.79	0.70
0.3778	158	0.84	0.75

Table III. CPU Time in Microseconds As a Function of p_{nl} at Various β for the One-Pivot Algorithm

β	$\langle N \rangle$	$p_{nl}=0.01$	$=0.05$	$=0.10$	$=0.25$	X
0.3690	20.3	46	55	68	102	411
0.3728	33.2	50	65	82	135	582
0.3744	43.9	52	71	95	161	678
0.3760	65.4	56	83	121	227	889
0.3771	102	61	104	167	352	1618
0.3778	158	74	140	237	473	2278

$s(j)$ along the walk.[18] However, for the modest values of N considered here, this modified algorithm is probably not advantageous.

We report in Tables III and IV the computer time on a VAX 8650 (running VMS Fortran) for a Monte Carlo step as a function of p_{nl} for the 1-pivot and 2-pivot algorithms at various values of $\langle N \rangle$. Clearly the local

[18] Recall how the proposed walk ω' is constructed if $s(i)$ precedes $s(j)$ along the walk: the algorithm works outward starting at the pivot points $s(i)$ and $s(j)$, which stay fixed; the steps of the walk preceding $s(i)$ are rewritten in place, as are the steps following $s(j)$; while the steps following $s(i)$ are overwritten with those preceding $s(j)$, and vice versa. All this is carried out using the pointers $p^{\pm}$. The result is $\omega' \equiv \omega^{0,i} \circ I\omega^{i,j} \circ \omega^{j,N}$. (Here we suppress, for notational simplicity, the distinction between indices along the walk and those in the linear list.) Now, what happens if we apply this same algorithm, but it turns out that $s(i)$ in fact *follows* $s(j)$? It is not hard to see that the algorithm constructs a well-defined walk ω' whose sequence of steps is exactly that of $I\omega^{j,N} \circ \omega^{i,j} \circ I\omega^{0,i}$, but which runs from $s(i)+s(j)-x$ to $s(i)+s(j)$ rather than from 0 to x. In this case it is easy to carry out at the end an overall inversion and translation to the origin, to produce the desired walk ω'. This latter operation takes a time of order N, but is performed only for *successful* nonlocal moves.

Table IV. CPU Time in Microseconds As a Function of p_{nl} at Various β for the Two-Pivot Algorithm

β	$\langle N \rangle$	$p_{nl}=0.01$	$=0.05$	$=0.075$	$=0.10$	$=0.15$	X
0.3690	20.3	47	53	56	60	66	420
0.3728	33.2	49	60	67	75	89	640
0.3744	43.9	52	66	—	84	—	735
0.3760	65.4	53	75	84	94	123	856
0.3771	102	58	91	110	123	164	1163
0.3778	158	62	120	144	167	248	1792

upgradings take a time which does not depend on the length of the walk; hence the mean time per iteration can be written as

$$T \approx (1 - p_{nl}) T_{\text{loc}} + p_{nl,\text{eff}} X(\langle N \rangle) \tag{3.46}$$

where T_{loc} is the computer time for the local (BFACF) moves, and $X(\langle N \rangle)$ is the mean time for a nonlocal move that is not immediately rejected. From this relation we determine experimentally the function $X(\langle N \rangle)$. We find that $X(\langle N \rangle)$ grows a bit slower than linearly in $\langle N \rangle$, for the values of $\langle N \rangle$ considered here; a reasonable fit is $X \sim \langle N \rangle^{0.8 \pm 0.2}$. This exponent is roughly comparable to the exponent 0.81 obtained by Madras and Sokal[23] for the pivot algorithm for walks of *fixed* length and *free* endpoints.

4. NUMERICAL RESULTS

We performed runs of both the 1-pivot and 2-pivot algorithms, on two-dimensional SAWs with fixed endpoint $|x| = 1$, at a sequence of values of β yielding average walk lengths $\langle N \rangle$ ranging from ≈ 20 to ≈ 160. At each β, we tried values of p_{nl} ranging between 0.01 and 0.15–0.25. We also did a few runs at $p_{nl} = 0.50$ and 0.75 for the smaller values of β. Tables V and VI show the parameters of our principal runs. In all cases we took data once every $\approx \tau_{\text{int},N}/10$ Monte Carlo steps, and the run lengths were typically a few thousand times $\tau_{\text{int},N}$. We also quote data for $p_{nl} = 0$ (pure BFACF algorithm) taken from ref. 10, to which we have added a run of 1.8×10^{10} iterations at $\beta = 0.3771$ (taking data once every 1.8×10^4 iterations).

We also made some runs of the 2-pivot algorithm in which each nonlocal move consisted of 10 "hits," for the purpose of testing whether the

Table V. Parameters of Our Runs for the One-Pivot Algorithm[a]

β	Data-taking interval	Run length			
		$p_{nl} = 0.01$	$p_{nl} = 0.05$	$p_{nl} = 0.10$	$p_{nl} = 0.25$
0.3690	1×10^3	4.0×10^8	3.0×10^8	2.0×10^8	2.0×10^8
0.3728	5×10^3	5.0×10^8	5.5×10^8	5.0×10^8	2.5×10^8
0.3744	1×10^4	1.5×10^9	1.0×10^9	1.0×10^9	8.65×10^8
0.3760	2×10^4	2.6×10^9	2.2×10^9	1.8×10^9	7.7×10^8
0.3771	4×10^4	5.2×10^9	4.0×10^9	4.0×10^9	1.16×10^9
0.3778	8×10^4	7.2×10^9	4.0×10^9	1.2×10^9	1.2×10^9

[a] All times are measured in MC steps.

Table VI. Parameters of Our Runs for the Two-Pivot Algorithm[a]

β	Data-taking interval	Run length			
		$p_{nl}=0.01$	$p_{nl}=0.05$	$p_{nl}=0.075$	$p_{nl}=0.15$
0.3690	1×10^3	5.0×10^8	—	4.0×10^8	4.0×10^8
0.3728	5×10^3	5.0×10^8	—	2.5×10^8	2.5×10^8
0.3744	1×10^4	5.0×10^8	5.0×10^8	—	—
0.3760	2×10^4	1.8×10^9	1.2×10^9	8.0×10^8	1.2×10^9
0.3771	4×10^4	1.2×10^{10}	2.8×10^9	2.8×10^9	1.6×10^9
0.3778	8×10^4	2.4×10^9	4.0×10^9	4.0×10^9	1.5×10^9

[a] All times are measured in MC steps.

Table VII. Parameters of Our Runs for the Two-Pivot Algorithm with Ten "Hits"[a]

β	Data-taking interval	Run length		
		$p_{nl}=0.001$	$p_{nl}=0.005$	$p_{nl}=0.010$
0.3690	1×10^3	1.5×10^8	1.5×10^8	1.0×10^9
0.3728	5×10^3	5.0×10^8	5.0×10^8	5.0×10^8
0.3744	1×10^4	1.0×10^9	9.0×10^8	9.0×10^8
0.3760	2×10^4	3.2×10^9	8.0×10^8	8.0×10^8
0.3771	4×10^4	6.0×10^9	5.2×10^9	2.7×10^9
0.3778	8×10^4	2.4×10^9	2.4×10^9	2.4×10^9

[a] All times are measured in MC steps.

Table VIII. Best Estimates of Static Means of N (Number of Bonds in Walk), N^2, S_N^2 (Squared Radius of Gyration), and $|\mathcal{A}|$ (Area Enclosed by Walk)[a]

β	$\langle N\rangle$	$\langle N^2\rangle$	$\langle S_N^2\rangle$	$\langle\lvert\mathcal{A}\rvert\rangle$
0.3690	20.3 (0.2)	1189 (23)	8.3 (0.2)	20.7 (0.5)
0.3728	33.0 (0.3)	3197 (111)	17.2 (0.4)	43 (1)
0.3744	44.2 (0.5)	5588 (204)	26.1 (0.6)	60 (1)
0.3760	65.3 (0.7)	12135 (360)	46 (1)	117 (2)
0.3771	102 (1)	29576 (879)	91 (2)	228 (5)
0.3778	158 (4)	73179 (3626)	177 (6)	445 (15)

[a] Estimates are weighted means combining data from all runs. Standard error is shown in parentheses.

nonlocal moves are efficient "randomizers" at fixed N (see Section 5 for further discussion). The parameters of these runs are shown in Table VII.

We have analyzed the data using standard procedures of statistical time-series analysis[60]; more details can be found in ref. 23, Appendix C. We used in all cases a self-consistent truncation window of width $5\tau_{\text{int},A}$.

In Table VIII we report our best estimates for the mean values of N, N^2, S_N^2, and $|\mathscr{A}|$ (see Section 2.1 for definitions), obtained by combining data from all runs.

In Tables IX–XI we report our estimates of the autocorrelation time $\tau_{\text{int},N}$ as a function of β and p_{nl}. For the 1-pivot algorithm, we also report estimates of $\tau_{\text{int},\mathscr{A}}$ (here $\mathscr{A}$ is the *signed* area). For the 2-pivot algorithm, $\tau_{\text{int},\mathscr{A}}$ is essentially zero.[19] All other observables show dynamic behavior that is qualitatively similar to that of N: the autocorrelation times of N^2, S_N^2, and $|\mathscr{A}|$ are in all cases roughly one-and-a-half times that of N (up to statistical error).

[19] This is easy to understand. For a *closed loop*, the mean value of $\mathscr{A}(t=1)$ conditional on $\omega(t=0)$ is zero by symmetry: each segment of the walk has the same probability of being chosen for inversion as the complementary segment, and the resulting walks are equivalent by overall inversion; in particular they have equal and opposite signed areas. Therefore the autocorrelation function of $\mathscr{A}$ vanishes at all nonzero time lags. For a walk with fixed $x \neq 0$, the autocorrelation time of $\mathscr{A}$ is not strictly zero, but it is negligible on the time scale we are looking at.

Table IX. Autocorrelation Times for the One-Pivot Algorithm[a]

β	$p_{nl}=0$	$p_{nl}=0.01$	$p_{nl}=0.05$	$p_{nl}=0.10$	$p_{nl}=0.25$
0.3690	5.7 (0.2)	4.3 (1.2)	3.1 (0.6)	2.4 (0.6)	2.3 (0.6)
		6.1 (2.0)	3.4 (0.8)	1.8 (0.4)	1.0 (0.2)
0.3728	23 (1)	11 (1)	8.1 (0.5)	7 (1)	5 (1)
		11 (1)	5.0 (0.2)	3.4 (0.2)	1.8 (0.1)
0.3744	50 (3)	27 (3)	16 (2)	15 (2)	10 (2)
			5.9 (0.4)	3.8 (0.2)	2.0 (0.1)
0.3760	190 (14)	64 (8)	39 (4)	29 (3)	32 (5)
		29 (3)	10.2 (0.4)	6.7 (0.4)	
0.3771	547 (86)	169 (27)	70 (8)	80 (10)	56 (8)
	592 (96)	46 (4)	10.8 (0.4)	8.4 (0.3)	
0.3778	—	464 (114)	216 (32)	168 (30)	120 (30)
		40 (2)	12.0 (0.3)	8.6 (0.5)	4.4 (0.4)

[a] First row is $\tau_{\text{int},N}$; second row is $\tau_{\text{int},\mathscr{A}}$. All times are measured in units of 10^4 MC steps. Standard error is shown in parentheses.

Table X. Autocorrelation Times $\tau_{\text{int},N}$ for the Two-Pivot Algorithm[a]

β	$p_{nl}=0$	$p_{nl}=0.01$	$p_{nl}=0.05$	$p_{nl}=0.075$	$p_{nl}=0.15$
0.3690	5.7 (0.2)	3.5 (0.5)	—	2.2 (0.2)	1.8 (0.2)
0.3728	23 (1)	14 (2)	—	10 (1)	7 (1)
0.3744	50 (3)	32 (8)	10 (2)	—	—
0.3760	190 (14)	32 (2)	24 (3)	21 (2)	22 (2)
0.3771	547 (86)	82 (8)	82 (12)	72 (5)	48 (8)
0.3778	—	328 (83)	198 (41)	95 (14)	123 (25)

[a] All times are measured in units of 10^4 MC steps. Standard error is shown in parentheses.

It can be proven (see Theorem A.5 in the Appendix) that, at fixed β, $1/(\tau_{\text{int},A}+1/2)$ is a *concave* function of p_{nl} for any observable A. This theorem provides a good check on our numerical data, and also signals which data points may be too high or too low due to statistical fluctuations. Our data show, in fact, a broad "plateau" in which $\tau_{\text{int},A}$ is reasonably close to its minimum value; this region ranges roughly between $p_{nl}=0.1$ and $p_{nl}=0.5$, and does not seem to change with β. For $p_{nl} \lesssim 0.1$, $\tau_{\text{int},A}$ at fixed β seems to grow like $1/p_{nl}^{\approx 0.3}$. On the other hand, if we fix p_{nl} and vary β, then $\tau_{\text{int},A}$ grows according to (3.27) with dynamic critical exponent

$$p_A = 2.0 \pm 0.2 \tag{4.1}$$

This should be compared with the exponent

$$p_A = 3.0 \pm 0.2 \tag{4.2}$$

for the pure BFACF algorithm ($p_{nl}=0$); see Fig. 7.

Table XI. Autocorrelation Times $\tau_{\text{int},N}$ for the Two-Pivot Algorithm with Ten "Hits"[a]

β	$p_{nl}=0$	$p_{nl}=0.001$	$p_{nl}=0.005$	$p_{nl}=0.010$
0.3690	5.7 (0.2)	3.5 (0.5)	2.5 (0.3)	1.9 (0.3)
0.3728	23 (1)	11 (1)	6.5 (1)	6.5 (1)
0.3744	50 (3)	23 (4)	13 (2)	10 (1)
0.3760	190 (14)	44 (4)	24 (4)	24 (4)
0.3771	547 (86)	106 (12)	80 (8)	66 (8)
0.3778	—	456 (88)	168 (48)	112 (16)

[a] All times are measured in units of 10^4 MC steps. Standard error is shown in parentheses.

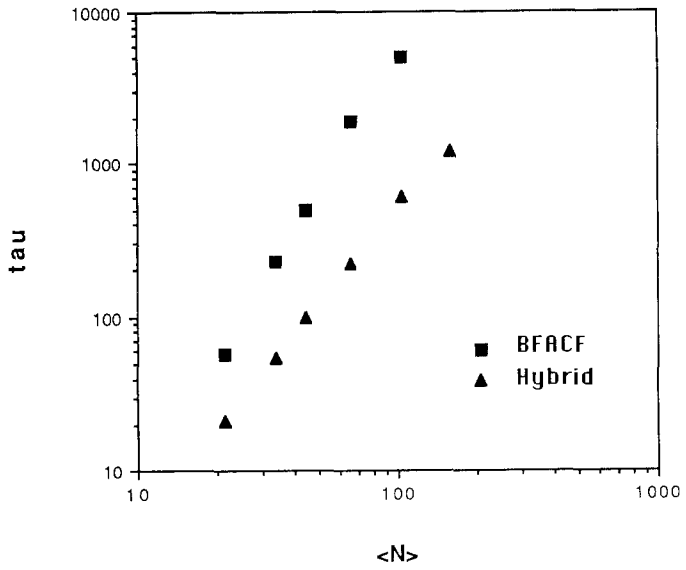


Fig. 7. Log–log plot of $\tau_{\text{int},N}$ as a function of $\langle N \rangle$ for $p_{nl}=0$ (pure BFACF algorithm) and $p_{nl}=0.25$ (1-pivot algorithm).

The exponent (4.1) is in fact the *best* that can be obtained by *any* variable-N algorithm that makes bounded changes in N (in our case $|\Delta N| \leqslant 2$) in each move. This is because the observable N will, in the best case, perform a random walk on the nonnegative integers, leading to τ_{exp}, $\tau_{\text{int},N} \sim \langle N \rangle^2$. A rigorous proof of this theorem can be found in the Appendix: see Theorems A.6 and A.7 and Example 2 following the latter. (Let us remark that *fixed-N* algorithms, such as the pivot algorithm for free-endpoint SAWs, can in some cases achieve a dynamic critical exponent that is nearly zero.[23])

In Fig. 8 we plot $\tau_{\text{int},N}/\langle N \rangle^2$ versus $p_{nl,\text{eff}}$ for various values of β. The data points fall roughly onto a single scaling curve, i.e.,

$$\tau_{\text{int},N}(\beta, p_{nl}) \approx \langle N \rangle^2 \mathscr{F}(p_{nl}) \tag{4.3}$$

as $\beta \uparrow \beta_c$, for some scaling function $\mathscr{F}$. As noted above, $\mathscr{F}(p_{nl}) \sim 1/p_{nl}^{\approx 0.3}$ as $p_{nl} \downarrow 0$.

Up to now we have used as a unit of time the number of steps in the Monte Carlo procedure, but clearly we are more interested in computer-time (CPU) units. We want to know how much time our computer must run to produce results with the chosen statistical error bars—after all, it is this time which has an influence on our budget! While in the BFACF

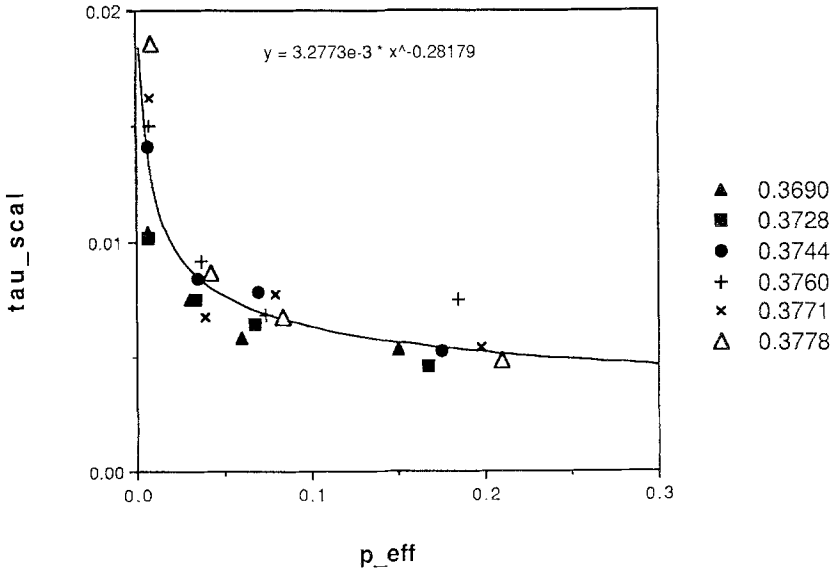


Fig. 8. Plot of $\tau_{\text{int},N}/\langle N\rangle^2$ versus p_{nl} for various values of β (1-pivot algorithm). The points fall roughly onto a scaling curve $\mathscr{F}(p_{nl})$.

algorithm the "physical" and CPU units of time are in a constant ratio, this is not the case for the hybrid algorithm, because the computational complexity of the nonlocal moves increases with $\langle N\rangle$. At fixed β, we have found (see Section 3.3) that the CPU time per Monte Carlo step behaves according to (3.46), where empirically

$$X \sim \langle N\rangle^{0.8 \pm 0.2} \tag{4.4}$$

Using the relation

$$\tau_{\text{phys}} \sim \langle N\rangle^{\approx 2}/p_{nl}^{\approx 0.3} \tag{4.5}$$

together with (3.46) and (4.4), we obtain

$$\tau_{\text{CPU}} \sim \langle N\rangle^{\approx 2}[(1-p_{nl})T_{\text{loc}} + p_{nl}T'\langle N\rangle^{\approx 0.8}]/p_{nl}^{\approx 0.3} \tag{4.6}$$

Elementary calculation then shows that the optimum value of p_{nl} scales as

$$p_{nl,\text{opt}} \sim 1/\langle N\rangle^{\approx 0.8} \tag{4.7}$$

—that is, we should spend a roughly equal amount of CPU time on local

and nonlocal moves—and the resulting CPU time per "effectively independent data point" scales as

$$\tau_{\text{CPU}} \sim \langle N \rangle^{\approx 2.3} \tag{4.8}$$

This is significantly better than the pure BFACF algorithm ($\tau_{\text{CPU}} \sim \langle N \rangle^{\approx 3.0}$).

5. CONCLUSIONS

Our initial attempt to understand the dynamic critical behavior of the hybrid BFACF/cut-and-paste algorithm was based on the idea that one or a few nonlocal moves would act as an efficient "randomizer" around the subspace of fixed N (and fixed number $N_1^{\pm}, N_2^{\pm}, \ldots, N_d^{\pm}$ of links in each direction). This idea is motivated by the behavior of the pivot algorithm for free-endpoint SAWs,[23] in which $\sim N^{\approx 0.2}$ attempted nonlocal moves are sufficient to equilibrate all global observables. If this conjecture were correct, then the local (BFACF) moves would carry out a random walk in N, leading to $\tau \sim \langle N \rangle^2$. One would further expect that this behavior could be achieved with a rather small p_{nl}, possibly of order $\langle N \rangle^{-2}$: for if a single nonlocal move were a good randomizer, then this randomization would need to be repeated only once every autocorrelation time of the combined algorithm, i.e., once every time $\sim \langle N \rangle^2$. That is, we would conjecture a scaling behavior of the form

$$\tau_{\text{int},N}(\beta, p_{nl}) \approx \langle N \rangle^2 \mathscr{F}(p_{nl} \langle N \rangle^r) \tag{5.1}$$

with $r \approx 2$. However, our numerical data do *not* support this conjecture: we find that a scaling form (5.1) does indeed hold, but with $r \approx 0$!

In order to obtain a better understanding of the effect of the nonlocal moves, we made some runs in which a single nonlocal move consists of *ten* 2-pivot inversions rather than one. If the 2-pivot inversions were indeed perfect randomizers around the given subspace, then ten hits would be no better than one. (In mathematical terms, the transition probability matrix for 2-pivot inversions would be approximately idempotent, $P_{\text{2-piv}}^n \approx P_{\text{2-piv}}$ for all $n > 1$.) However, our numerical data show that ten inversions *are* better than one. Indeed, the 10-hit algorithm with a given p_{nl} has essentially the same autocorrelation time as the 1-hit algorithm with $10 p_{nl}$. What really matters, therefore, appears to be the total number of nonlocal "hits"; it is essentially irrelevant whether they are performed one after another or interspersed with the local moves. We conclude that the matrix $P_{\text{2-piv}}$ is very *far* from being idempotent.

Therefore, our initial conjecture that a small (of order $\langle N \rangle^{\approx 0}$)

number of nonlocal moves would act as an essentially perfect randomizer at fixed N appears to be far from correct.[20] On the other hand, if we use a larger number of nonlocal moves—p_{nl} *constant* as $\langle N \rangle \to \infty$, leading to $\sim \langle N \rangle^2$ nonlocal moves per autocorrelation time of the combined algorithm—then the hybrid algorithm *does* achieve the predicted $\tau \sim \langle N \rangle^2$ behavior. As the number of nonlocal moves is reduced, the performance of the hybrid algorithm deteriorates, but only rather slowly ($\tau \sim 1/p_{nl}^{\approx 0.3}$). Indeed, this latter exponent can be predicted by the following heuristic argument: We known that the pure BFACF algorithm has an autocorrelation time $\tau_{\text{int},N} \sim \langle N \rangle^p$ with $p \approx 3.0$. Clearly, if we were to make only one nonlocal move per autocorrelation time of the pure BFACF algorithm, then that nonlocal move would be essentially redundant (i.e., the autocorrelation time would not change much); while if we were to make nonlocal moves significantly more often than this (i.e., some power of $\langle N \rangle$ more often), then one might expect the dynamic critical exponent to be reduced. Suppose, then, that the autocorrelation time $\tau_{\text{int},N}$ of the hybrid algorithm has a scaling behavior (5.1) with $\mathscr{F}(y) \sim y^{-s}$ for some exponent s. Setting $p_{nl} \sim \langle N \rangle^{-p}$ in (5.1) and insisting that $\tau_{\text{int},N} \sim \langle N \rangle^p$, we obtain $\langle N \rangle^{2-rs+ps} \sim \langle N \rangle^p$; hence $s = (p-2)/(p-r)$. If we now insert $p \approx 3$ and $r \approx 0$, we obtain $s \approx 0.3$.

Our understanding of the behavior of the hybrid algorithm is therefore self-consistent but still somewhat incomplete. One would like to understand *why* the nonlocal moves are less efficient randomizers than we had initially thought—that is, one would like to understand why r is approximately zero rather than approximately 2. We suspect that this is related to the presence of large but *not* precisely rectangular configurations (of area $\mathscr{A} \sim N^{1+\varepsilon}$ with $0 < \varepsilon < 1$) that require *many* (of order N^{δ}) nonlocal moves in order to be reduced to a walk with area $\sim N$.

In any case, the nonlocal moves are sufficiently powerful that even with p_{nl} as small as $1/\langle N \rangle^{\approx 0.8}$—the maximum we can afford in terms of computer time—the autocorrelation time is still only $\sim \langle N \rangle^{\approx 2.3}$. This exponent is not far from the "ideal" exponent 2, and is considerably lower than the BFACF exponent ≈ 3. In practice, this means that already at $\langle N \rangle \approx 100$ we find a physical (resp. CPU) autocorrelation time for the hybrid algorithm with $p_{nl} = 0.05$ that is a factor 6 (resp. 4) smaller than that of the pure BFACF algorithm. Moreover, for larger $\langle N \rangle$ the gain will improve as $\sim \langle N \rangle^{\approx 0.7}$. The hybrid algorithm provides, therefore, a substantial improvement over previous algorithms for fixed-endpoint SAWs.

[20] Vaguely similar behavior occurs in the Swendsen–Wang[61] algorithm for the Potts model, which has nontrivial critical slowing-down in spite of the nonlocality of the algorithm.[62]

APPENDIX. SOME RIGOROUS BOUNDS

In this Appendix we prove some rigorous bounds on the autocorrelation times $\tau_{\exp}$ and $\tau_{\text{int},A}$ of reversible Markov chains. These theorems fall into three categories:

1. Comparison theorems (Theorems A.1–A.3).
2. General properties of "hybrid" Monte Carlo algorithms $P_\lambda = (1-\lambda)P_0 + \lambda P_1$ (Theorems A.4–A.5).
3. Bounds based on random-walk ideas (Theorems A.6–A.7).

For notational simplicity, we consider Markov chains whose state space S is *discrete* (i.e., finite or countably infinite); however, all our theorems and proofs carry over immediately to the case of a general (measurable) state space, with only minor notational alterations (replacing matrices by kernels, and sums by integrals).

Let, therefore, $P = \{p_{xy}\}_{x,y \in S}$ be an *irreducible* transition probability matrix on S, and assume that P has a stationary probability distribution π (necessarily unique). Let $l^2(\pi)$ be the space of complex-valued functions on S that are square-integrable with respect to π. This is a Hilbert space with inner product

$$(f, g) \equiv \sum_x f(x)^* g(x) \tag{A.1}$$

and norm $\|f\| \equiv (f, f)^{1/2}$. The matrix P acts naturally on $l^2(\pi)$ by

$$(Pf)(x) = \sum_y p_{xy} f(y) \tag{A.2}$$

It is not hard to show that P is a contraction on $l^2(\pi)$, i.e., $\|Pf\| \leqslant \|f\|$ for all $f \in l^2(\pi)$; in particular, the spectrum of P lies in the closed unit disk. The constant function $\mathbf{1}$ is an eigenvector of P (and of its adjoint P^*) with eigenvalue 1, and this eigenvalue is simple. Let R be the spectral radius of P acting on the orthogonal complement of the constant functions:

$$R \equiv \inf\{r: \operatorname{spec}(P \restriction \mathbf{1}^\perp) \subset \{\lambda: |\lambda| \leqslant r\}\} \tag{A.3}$$

Then it can be shown[51] that $R = e^{-1/\tau_{\exp}}$.

It is convenient to introduce the operator Π defined by

$$(\Pi f)(x) \equiv \sum_y \pi_y f(y) \qquad \text{for all } x \tag{A.4}$$

Clearly $\Pi f = (\mathbf{1}, f)\mathbf{1} = \langle f \rangle_\pi \mathbf{1}$, so Π is the orthogonal projection in $l^2(\pi)$ onto the constant functions. Therefore, $\Pi^\perp \equiv I - \Pi$ is the orthogonal

projection in $l^2(\pi)$ onto the orthogonal complement of the constant functions, i.e., the functions f having $\langle f \rangle_\pi = 0$. It is easy to see that $\Pi P = P\Pi = \Pi$.

From now on we restrict attention to Markov chains that are *reversible* (i.e., satisfy *detailed balance*) with respect to π [cf. (2.36)]. This condition is equivalent to the self-adjointness of P on $l^2(\pi)$. The spectrum of P therefore lies in the interval $[-1, 1]$; there is a simple eigenvalue at 1 with eigenvector equal to the constant function, and we wish to know how close the rest of the spectrum gets to 1. We define therefore the *spectral gap* (or *mass gap*)

$$m \equiv 1 - \sup \operatorname{spec}(P \restriction \mathbf{1}^\perp) \tag{A.5}$$

By the Rayleigh–Ritz principle,

$$m = \sup_{\substack{f \in \mathbf{1}^\perp \\ f \neq 0}} \frac{(f, (I-P)f)}{(f, f)} \tag{A.6}$$

We also define the *modified autocorrelation time*

$$\tau'_{\exp} \equiv \begin{cases} -1/\log(1-m) & \text{if } m < 1 \\ 0 & \text{if } m \geqslant 1 \end{cases} \tag{A.7}$$

$\tau'_{\exp}$ is very much like $\tau_{\exp}$ except that it is controlled by the spectrum of P near $+1$ only, while $\tau_{\exp}$ is controlled by the spectrum near both $+1$ and -1. For most purposes in Monte Carlo work, only the spectrum near $+1$ matters (see ref. 51, note 8, for further discussion).

Next we introduce the *limiting covariance operator*[21]

$$C \equiv \begin{cases} (I+P)/(I-P) & \text{on } \mathbf{1}^\perp \\ 0 & \text{on } \mathbf{1} \end{cases} \tag{A.8}$$

C is self-adjoint and positive-semidefinite. (If $m = 0$, then C is an unbounded operator, but it is in any case densely defined.) C is called the "limiting covariance operator" because of the following fact:

Proposition A.1. (a) Let $f \in l^2(\pi)$, and let $X_0, X_1, \ldots$ be the successive states of the Markov chain P started in its stationary distribution π. Then the limiting variance of the sample mean of f is given by

$$\lim_{n \to \infty} n \operatorname{var}\left(\frac{1}{n} \sum_{t=1}^{n} f(X_t)\right) = \begin{cases} (f, Cf) & \text{if } f \in \mathcal{Q}(C) \\ +\infty & \text{if } f \notin \mathcal{Q}(C) \end{cases} \tag{A.9}$$

[21] For Markov chains with *finite* state space, the limiting covariance matrix is discussed in ref. 53, Sections 4.6 and 5.1, and ref. 54, Sections 4.3.5 and 4.4.4.

[Here $\mathcal{Q}(C)$ is the quadratic form domain of C; see ref. 63, Section VIII.6, or ref. 64, Chapter 6, for a definition and discussion.]

(b) Let $f, g \in \mathcal{Q}(C)$. Then the limiting covariance of the sample means of f and g is given by

$$\lim_{n \to \infty} n \operatorname{cov}\left(\frac{1}{n}\sum_{t=1}^{n} f(X_t), \frac{1}{n}\sum_{t=1}^{n} g(X_t)\right) = (f, Cg) \tag{A.10}$$

Proof. By time-translation invariance (stationarity),

$$\begin{aligned} & n \operatorname{cov}\left(\frac{1}{n}\sum_{t=1}^{n} f(X_t), \frac{1}{n}\sum_{t=1}^{n} g(X_t)\right) \\ & \equiv \frac{1}{n}\sum_{s,t=1}^{n} \operatorname{cov}(f(X_s), g(X_t)) \\ & = \sum_{t=-(n-1)}^{n-1} \left(1 - \frac{|t|}{n}\right) \operatorname{cov}(f(X_0), g(X_t)) \\ & = \sum_{t=-(n-1)}^{n-1} \left(1 - \frac{|t|}{n}\right) (\Pi^{\perp} f, P^{|t|} \Pi^{\perp} g) \end{aligned} \tag{A.11}$$

where we have used the self-adjointness of P to handle the terms with $t < 0$. The sum (A.11) is a Cesaro sum. To prove that it converges to the claimed limit, we use the spectral theorem: Let $dE(\lambda)$ be the spectral measure for the operator P, and let $d\mu_{fg}(\lambda) = (\Pi^{\perp} f, dE(\lambda)\, \Pi^{\perp} g)$. Consider first the case $f, g \in \mathcal{Q}(C)$. The $d\mu_{fg}$ is a finite complex measure on the interval $[-1, 1)$ satisfying

$$\int_{-1}^{1} \frac{1+\lambda}{1-\lambda}\, d|\mu_{fg}|(\lambda) < \infty \tag{A.12}$$

Now (A.11) is equal to

$$\sum_{t=-(n-1)}^{n-1} \left(1 - \frac{|t|}{n}\right) \int_{-1}^{1} \lambda^{|t|}\, d\mu_{fg}(\lambda) = \int_{-1}^{1} \left[\frac{1+\lambda}{1-\lambda} - \frac{2\lambda(1-\lambda^n)}{n(1-\lambda)^2}\right] d\mu_{fg}(\lambda) \tag{A.13}$$

It is not hard to see that the brackets are bounded between 0 and $(1+\lambda)/(1-\lambda) + (4/n)$. Moreover, for each fixed $\lambda \in [-1, 1)$, the brackets tend to $(1+\lambda)/(1-\lambda)$ as $n \to \infty$. It therefore follows from (A.12) and the Lebesgue dominated convergence theorem that (A.11) tends to

$$\int_{-1}^{1} \frac{1+\lambda}{1-\lambda}\, d\mu_{fg}(\lambda) \tag{A.14}$$

as $n \to \infty$. But this equals (f, Cg).

If $f = g \notin \mathcal{D}(C)$, then $d\mu_{ff}$ is a finite positive measure on $[-1, 1)$ satisfying

$$\int_{-1}^{1} \frac{1+\lambda}{1-\lambda} d\mu_{ff}(\lambda) = +\infty \tag{A.15}$$

But then the bounds proven earlier imply, by Fatou's lemma, that (A.11) tends to $+\infty$. ∎

Rephrasing this theorem in terms of autocorrelation times, we have

$$\tau_{\text{int},f} = \frac{1}{2} \frac{(f, Cf)}{(f, \Pi^{\perp} f)} \tag{A.16}$$

for any nonconstant $f \in \mathcal{D}(C)$ [and $\tau_{\text{int},f} = +\infty$ if $f \notin \mathcal{D}(C)$].

Let us recall, finally, that the autocorrelation function of an observable $f \in l^2(\pi)$ can be written as

$$\begin{aligned} C_{ff}(t) &= (f, (P^{|t|} - \Pi) f) \\ &= (\Pi^{\perp} f, P^{|t|} \Pi^{\perp} f) \end{aligned} \tag{A.17}$$

By the spectral theorem, this implies

$$C_{ff}(t) = \int_{-1}^{1} \lambda^{|t|} \, d\sigma_{ff}(\lambda) \tag{A.18}$$

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

$$\begin{aligned} \tau_{\text{int},f} &= \frac{1}{2} \frac{\int_{-1}^{1} [(1+\lambda)/(1-\lambda)] \, d\sigma_{ff}(\lambda)}{\int_{-1}^{1} d\sigma_{ff}(\lambda)} \\ &\geqslant \frac{1}{2} \frac{1+\rho_{ff}(1)}{1-\rho_{ff}(1)} \end{aligned} \tag{A.19}$$

by Jensen's inequality [since the function $\lambda \mapsto (1+\lambda)/(1-\lambda)$ is convex].[22] One method for proving lower bounds on $\tau_{\text{int},f}$ (and hence also on τ_{exp}) is to compute an explicit upper bound on the Rayleigh quotient

$$\frac{(f, (I-P)f)}{(f, (I-\Pi)f)} = \frac{C_{ff}(0) - C_{ff}(1)}{C_{ff}(0)} = 1 - \rho_{ff}(1) \tag{A.20}$$

We will use this method in the proof of Theorem A.7 below.

[22] It also follows from (A.18) that $\rho_{ff}(t) \geqslant \rho_{ff}(1)^{|t|}$ for *even* values of t. Moreover, this holds for odd values of t if $d\sigma_{ff}$ is supported on $\lambda \geqslant 0$ (though not necessarily otherwise). Therefore, the dcecay as $t \to \infty$ of $\rho_{ff}(t)$ is also bounded below in terms of $\rho_{ff}(1)$.

We now discuss the comparison of two reversible Markov chains P and P' that both satisfy detailed balance for the same stationary distribution π. Let us first recall that if A and B are bounded self-adjoint operators on a Hilbert space $\mathscr{H}$, we write $A \leqslant B$ in case $(f, Af) \leqslant (f, Bf)$ for all $f \in \mathscr{H}$. If A and B are unbounded but positive-semidefinite, $A \leqslant B$ has the same meaning provided we make the convention that $(f, Af) = +\infty$ whenever $f \notin \mathscr{Q}(A)$, and likewise for B.

So let P and P' be two transition probabilities that satisfy detailed balance for π, and let C and C' be the corresponding limiting covariance operators. We then have the following easy comparison theorems:

Theorem A.1. Assume that $P \leqslant P'$. Then $\tau'_{\exp}(P) \leqslant \tau'_{\exp}(P')$. More generally, assume that $(I-P) \geqslant \alpha(I-P')$ for some $\alpha > 0$. Then $m(P) \geqslant \alpha m(P')$; and if $\alpha \geqslant 1$, then $\tau'_{\exp}(P) \leqslant \alpha^{-1}\tau'_{\exp}(P')$.

Proof. From the Rayleigh–Ritz principle (A.6), it follows that $m(P) \geqslant \alpha m(P')$. The final claim follows from the fact that the function $m \mapsto -\log(1-m)$ is increasing and convex. ∎

Theorem A.2. (a) $P \leqslant P'$ if and only if $C \leqslant C'$.

(b) More generally, for any $\alpha > 0$, $(I-P) \geqslant \alpha(I-P')$ if and only if $(C+I) \leqslant \alpha^{-1}(C'+I)$ on $\mathbf{1}^{\perp}$. {In terms of autocorrelation times, this says that $\tau_{\mathrm{int},f}(P) + 1/2 \leqslant \alpha^{-1}[\tau_{\mathrm{int},f} + 1/2]$ for all observables $f \in l^2(\pi)$.}

Proof. On $\mathbf{1}^{\perp}$, $C + I = 2/(I-P)$ and $C' + I = 2/(I-P')$. The theorem then follows from the well-known fact that $0 \leqslant A \leqslant B$ implies $0 \leqslant A^{-1} \leqslant B^{-1}$.[23] ∎

Theorem A.3. Assume that $p_{xy} \geqslant p'_{xy}$ for all $x \neq y$. Then $P \leqslant P'$. More generally, assume that $p_{xy} \geqslant \alpha p'_{xy}$ for some $\alpha > 0$. Then $(I-P) \geqslant \alpha(I-P')$.

Proof. An easy computation shows that

$$(f, (I-P)f) = \tfrac{1}{2} \sum_{x,y} \pi_x p_{xy} |f(x) - f(y)|^2 \tag{A.21}$$

and likewise for P'. If $p_{xy} \geqslant \alpha p'_{xy}$ for all $x \neq y$, then $(f, (I-P)f) \geqslant \alpha(f, (I-P')f)$. ∎

[23] This result is a special case of Löwner's theorem.[66,67] On the other hand, there is a very elegant elementary proof[68]: noting that $(f, A^{-1}f) = \sup_g [2(g,f) - (g, Ag)]$ and likewise for B, the result follows immediately.

Remarks. 1. Combining Theorems A.2 and A.3 for $\alpha = 1$, we see that $p_{xy} \geqslant p'_{xy}$ for all $x \neq y$ implies $C \leqslant C'$, hence $\tau_{\text{int},f}(P) \leqslant \tau_{\text{inf},f}(P')$ for all observables f. This result was first proven by Peskun.[65]

2. The intuition behind Theorem A.3 (as combined with Theorems A.1 and A.2) is very natural: if P makes more transitions than P', it should equilibrate faster. However, it should be emphasized that this intuition is valid in general only for *reversible* Markov chains. Consider, for example, an Ising model at infinite temperature. If the sites are updated sequentially using a single-spin-flip Metropolis algorithm, then every spin-flip proposal is accepted, and the spins oscillate up and down deterministically; in particular, the Markov chain is nonergodic, so the algorithm *never* reaches equilibrium (1 is an eigenvalue of $P \restriction \mathbf{1}^\perp$). By contrast, if the single-spin-flip heat-bath algorithm were used, the acceptance probability would be 1/2, and the Markov chain would be ergodic and aperiodic (the spectrum of $P \restriction \mathbf{1}^\perp$ would be contained strictly inside the unit circle). This example does not contradict Theorem A.3, because the heat-bath algorithm with *sequential* site updating is not reversible. (It is a product of single-site updates, each one of which satisfies detailed balance, but the product does not, because the factors are noncommuting.) If *random* rather than sequential site updating were used, then both the Metropolis and heat-bath algorithms would be reversible, and Theorem A.3 would apply. We thank one of the referees for reminding us of this example.

Example. In the BFACF algorithm, the maximum possible values of $p(+2)$ and $p(-2)$ are given by (3.14) and (3.16), while the maximum possible value of $p(0)$ is 1/2. Therefore, the BFACF algorithm with the choices (3.14)–(3.16) satisfies the hypotheses of Theorem A.3 with $\alpha = (1+\beta^2)/[1+(2d-3)\beta^2]$ with respect to any other BFACF algorithm.

The next theorems concern "hybrid" algorithms of the form $P = P_\lambda \equiv (1-\lambda)P_0 + \lambda P_1$ $(0 \leqslant \lambda \leqslant 1)$, where P_0 and P_1 are transition matrices satisfying detailed balance for the same distribution π.

Theorem A.4. $m(P_\lambda)$ is a *concave* function of λ.

Proof. This is a well-known consequence of the Rayleigh–Ritz formula (A.6): the point is that if $\lambda_1 \leqslant \lambda_2 \leqslant \lambda_3$, then any trial function f for P_{λ_2} can also be used for P_{λ_1} and P_{λ_3}. ∎

A much deeper theorem is the following:

Theorem A.5. For each nonconstant $f \in l^2(\pi)$, $[\tau_{\text{int},f}(P_\lambda) + 1/2]^{-1}$ is a *concave* function of λ.

The proof of Theorem A.5 is based on a beautiful identity and inequality for "parallel addition" of self-adjoint operators due to Anderson *et al.*[69,70]:

Proposition A.2. Let A and B be positive-definite self-adjoint operators on a Hilbert space $\mathscr{H}$. Then

$$(f, [A^{-1}+B^{-1}]^{-1}f) = \inf_{\substack{g,h\in\mathscr{H} \\ g+h=f}} [(g, Ag)+(h, Bh)] \tag{A.22}$$

Sketch of Proof. The motivation behind this lemma is to think of A and B as multiport resistor networks and f as a set of input currents. Then (A.22) is a multiport generalization of the usual parallel-resistor formula: it says that the current divides between A and B so as to minimize the total power dissipation. In the finite-dimensional case, the proof is very simple (ref. 69, Lemma 18): Let $g_0 = (A+B)^{-1}Bf$. Then, for any $g, h\in\mathscr{H}$ with $g+h=f$, we have

$$(g, Ag)+(h, Bh) = (f, [A^{-1}+B^{-1}]^{-1}f)+(g-g_0, (A+B)(g-g_0)) \tag{A.23}$$

It follows that (A.22) holds, with equality when (and only when) $g=g_0$. The infinite-dimensional case is similar, but involves more technicalities, because the operators A^{-1} and B^{-1} might be unbounded (ref. 70, Theorems 8 and 9). ▮

Corollary A.1. Let A and B be positive-definite self-adjoint operators on a Hilbert space $\mathscr{H}$. Then

$$(f, [A^{-1}+B^{-1}]^{-1}f) \leqslant [(f, Af)^{-1}+(f, Bf)^{-1}]^{-1} \tag{A.24}$$

Proof. Set

$$g = \frac{(f, Bf)}{(f, (A+B)f)} f$$

in Proposition A.2. (Among choices of the form $g=\alpha f$, this is the optimal one.) ▮

Proof of Theorem A.5. We apply Corollary A.1 to the Hilbert space $\mathscr{H} = \mathbf{1}^{\perp}$, with $A = (1-\lambda)^{-1}(I-P_0)^{-1}$ and $B = \lambda^{-1}(I-P_1)^{-1}$. We conclude that

$$\begin{aligned}(f, [(1-\lambda)(I-P_0)+\lambda(I-P_1)]^{-1}f)^{-1} \\ \geqslant (1-\lambda)(f, (I-P_0)^{-1}f)^{-1}+\lambda(f, (I-P_1)^{-1}f)^{-1}\end{aligned} \tag{A.25}$$

Translation this into the language of autocorrelation times, we get

$$[\tau_{\text{int},f}(P_\lambda)+\tfrac{1}{2}]^{-1} \geqslant (1-\lambda)[\tau_{\text{int},f}(P_0)+\tfrac{1}{2}]^{-1} + \lambda[\tau_{\text{int},f}(P_1)+\tfrac{1}{2}]^{-1} \quad \text{(A.26)}$$

Theorem A.5 follows easily from this inequality. ∎

One immediate consequence of Theorems A.4 and A.5 is that

$$m(P_\lambda) \geqslant \min(\lambda, 1-\lambda) \times \sup_{0 \leqslant \lambda' \leqslant 1} m(P_{\lambda'}) \quad \text{(A.27)}$$

$$[\tau_{\text{int},f}(P_\lambda)+\tfrac{1}{2}]^{-1} \geqslant \min(\lambda, 1-\lambda) \times \sup_{0 \leqslant \lambda' \leqslant 1} [\tau_{\text{int},f}(P_{\lambda'})+\tfrac{1}{2}]^{-1} \quad \text{(A.28)}$$

In particular, the hybrid algorithm with $\lambda = 1/2$ is never more than a factor of 2 worse than the algorithm with the "optimal" value of λ.

Finally, we present two theorems that give lower bounds on the autocorrelation time based on a random-walk intuition. The general setup is the following: Suppose that the state space S can be decomposed as $S = \bigcup_{n=0}^{\infty} S_n$ in such a way that $p_{xy} = 0$ whenever $x \in S_i$ and $y \in S_j$ with $|i-j| > 1$. Then we can define an *aggregated Markov chain* $\bar{P}$ with state space $\mathbf{Z}_+ = \{0, 1, 2, \ldots\}$ and transition probabilities

$$\bar{p}_{ij} \equiv \frac{\sum_{x \in S_i,\, y \in S_j} \pi_x p_{xy}}{\sum_{x \in S_i} \pi_x} \quad \text{(A.29)}$$

This is an irreducible reversible Markov chain on $\mathbf{Z}_+$ with invariant measure

$$\bar{\pi}_i \equiv \sum_{x \in S_i} \pi_x \quad \text{(A.30)}$$

Moreover, it can easily be shown (ref. 51, Section 4) that

$$\tau'_{\text{exp}}(\bar{P}) \leqslant \tau'_{\text{exp}}(P) \quad \text{(A.31)}$$

Therefore, in order to prove lower bounds on $\tau'_{\text{exp}}(P)$, it suffices to prove lower bounds on τ'_{exp} (i.e., upper bounds on the mass gap) for general random walks on $\mathbf{Z}_+$. The next theorem addresses this question:

Theorem A.6. Let $P = \{p_{ij}\}$ be the transition matrix for a Markov chain on $\mathbf{Z}_+$ with invariant probability measure π, and suppose that $p_{ij} = 0$ for $|i-j| > 1$.[24] Fix $0 \leqslant r \leqslant 1$, and suppose that for all $\varepsilon > 0$ and all $k < \infty$, there exists M such that

$$r - \varepsilon \leqslant \frac{\pi_{n+1}}{\pi_n} \leqslant [2 - (r-\varepsilon)^{1/2}]^2 \quad \text{(A.32)}$$

[24] Such a Markov chain is automatically reversible, since the state space has no cycles except self-loops.

for $n = M, M+1, \ldots, M+k$. Then

$$\text{inf essential spec}(I-P) \leqslant \frac{(1-\sqrt{r})^2}{1+r} \tag{A.33}$$

[In particular, this is an upper bound on $m \equiv \inf \text{spec}(I-P)$.]

Remarks. 1. The hypothesis holds, in particular, if $\lim_{n\to\infty}(\pi_{n+1}/\pi_n) = r$. But it is of course much weaker.

2. The bound is sharp for the random walk with constant inward drift on $\mathbf{Z}_+$.

Proof. Consider the trial function

$$f(n) = \pi_n^{1/2} \chi(M \leqslant n \leqslant N) \tag{A.34}$$

Then, by (A.21), we have

$$\begin{aligned}
&(f, (I-P)f) \\
&\quad = \pi_{M-1} p_{M-1,M} \pi_M^{-1} + \pi_N p_{N,N+1} \pi_N^{-1} + \sum_{n=M}^{N-1} \pi_n p_{n,n+1} (\pi_n^{-1/2} - \pi_{n+1}^{-1/2})^2 \\
&\quad = p_{M,M-1} + p_{N,N+1} + p_{M,M+1} \left[\left(\frac{\pi_M}{\pi_{M+1}} \right)^{1/2} - 1 \right]^2 \\
&\qquad + p_{N,N-1} \left[1 - \left(\frac{\pi_N}{\pi_{N-1}} \right)^{1/2} \right]^2 \\
&\qquad + \sum_{n=M+1}^{N-1} \left\{ \lambda p_{n,n+1} \left[\left(\frac{\pi_n}{\pi_{n+1}} \right)^{1/2} - 1 \right]^2 \right. \\
&\qquad\qquad \left. + (1-\lambda) p_{n,n-1} \left[1 - \left(\frac{\pi_n}{\pi_{n-1}} \right)^{1/2} \right]^2 \right\}
\end{aligned} \tag{A.35}$$

for any $0 \leqslant \lambda \leqslant 1$. (Here we have repeatedly used the reversibility relation $\pi_i p_{ij} = \pi_j p_{ji}$.) Now, by hypothesis, there are disjoint intervals $[M_1, N_1]$, $[M_2, N_2]$,... with $\lim_{j\to\infty}(N_j - M_j) = \infty$ and $r - \varepsilon \leqslant \pi_{n+1}/\pi_n \leqslant [2-(r-\varepsilon)^{1/2}]^2$ for $M_j - 1 \leqslant n \leqslant N_j$. It follows that

$$\left| \left(\frac{\pi_n}{\pi_{n+1}} \right)^{1/2} - 1 \right| \leqslant \frac{1}{(r-\varepsilon)^{1/2}} - 1 \tag{A.36}$$

$$\left| 1 - \left(\frac{\pi_n}{\pi_{n-1}} \right)^{1/2} \right| \leqslant 1 - (r-\varepsilon)^{1/2} \tag{A.37}$$

for $M_j \leqslant n \leqslant N_j$. So choose $\lambda = (r-\varepsilon)/(1+r-\varepsilon)$. We then get, using $p_{n,n+1} + p_{n,n-1} \leqslant 1$,

$$(f, (I-P)f) \leqslant 3 + \left(\frac{1}{(r-\varepsilon)^{1/2}} - 1\right)^2 + (N-M-1)\frac{[1-(r-\varepsilon)^{1/2}]^2}{1+r-\varepsilon} \tag{A.38}$$

On the other hand,

$$(f, f) = \sum_{n=M}^{N} \pi_n \pi_n^{-1} = N - M + 1 \tag{A.39}$$

Thus, taking $j \to \infty$ and using $\lim_{j\to\infty}(N_j - M_j) = \infty$, we conclude that there is an infinite orthogonal family $\{f_j\}$ of trial functions that have Rayleigh quotients

$$\limsup_{j\to\infty} \frac{(f_j, (I-P)f_j)}{(f_j, f_j)} \leqslant \frac{[1-(r-\varepsilon)^{1/2}]^2}{1+r-\varepsilon} \tag{A.40}$$

By the min-max theorem, this proves that

$$\inf \text{essential spec}(I-P) \leqslant \frac{[1-(r-\varepsilon)^{1/2}]^2}{1+r-\varepsilon} \tag{A.41}$$

Since ε was arbitrary, the theorem is proven. ∎

Corollary A.2. Suppose that, in addition,

$$\sup_{n\geqslant 1} \frac{n\pi_n - (n-1)\, r\pi_{n-1}}{\pi_n} = C < \infty \tag{A.42}$$

Then

$$\inf \text{essential spec}(I-P) \leqslant C^2 \langle N \rangle^{-2} \tag{A.43}$$

where $\langle N \rangle \equiv \sum_{n=0}^{\infty} n\pi_n$.

Remark. If $\pi_n \sim r^n n^{\alpha-1}$ as $n \to \infty$, then $\lim_{n\to\infty}[n\pi_n - (n-1)\, r\pi_{n-1}]/\pi_n = \alpha$; so, provided that all $\pi_n > 0$ (i.e., the Markov chain is irreducible), it follows that $C < \infty$.

Proof. $\langle N \rangle = \sum_{n=1}^{\infty} n\pi_n$, so

$$\begin{aligned}(1-r)\langle N \rangle &= \sum_{n=1}^{\infty} [n\pi_n - (n-1)\, r\pi_{n-1}] \\ &\leqslant C \sum_{n=1}^{\infty} \pi_n \\ &\leqslant C\end{aligned} \tag{A.44}$$

Hence

$$\text{inf essential spec}(I-P) \leqslant \frac{(1-\sqrt{r})^2}{1+r} \leqslant (1-r)^2 \leqslant C^2 \langle N \rangle^{-2} \quad \blacksquare$$

(This corollary and its proof are essentially Corollary 4.1 of ref. 51.)

We can also prove a similar theorem for $\tau_{\text{int},f}$:

Theorem A.7. Let P be a reversible Markov chain with invariant probability measure π, and let $f \in l^2(\pi)$ be an observable whose maximum change in a single step of the Markov chain is bounded by $C < \infty$ (i.e., $|f(x)-f(y)| \leqslant C$ whenever $p_{xy} \neq 0$). Then

$$\tau_{\text{int},f} \geqslant \frac{2 \operatorname{var}_\pi(f)}{C^2} - \frac{1}{2} \tag{A.45}$$

Proof. Using (A.21), we see immediately that

$$(f, (I-P)f) \leqslant \frac{C^2}{2} \tag{A.46}$$

On the other hand,

$$(f, (I-\Pi)f) = \operatorname{var}_\pi(f) \equiv \langle f^2 \rangle_\pi - \langle f \rangle_\pi^2 \tag{A.47}$$

Therefore,

$$\rho_{ff}(1) = 1 - \frac{(f, (I-P)f)}{(f, (I-\Pi)f)} \geqslant 1 - \frac{C^2}{2 \operatorname{var}_\pi(f)} \tag{A.48}$$

which by (A.19) implies the claimed bound (A.45). $\blacksquare$

Examples. 1. In the BFACF algorithm with endpoint x, let $f(\omega) = \mathscr{A}(\omega, \omega')$ be the minimum surface area spanned by the union of ω and ω', where ω' is a fixed walk from 0 to x. (In the BFACF algorithm with $d=2$ and $|x|=1$, we could alternatively take f to be the signed area $\mathscr{A}$ or its absolute value $|\mathscr{A}|$.) Clearly $\mathscr{A}$ changes by at most one unit in a BFACF move, so Theorem A.7 implies that

$$\tau_{\text{int},\mathscr{A}} \geqslant 2 \operatorname{var}_\pi(\mathscr{A}) - 1/2 \tag{A.49}$$

and likewise for $|\mathscr{A}|$. Assuming that the full probability distribution of $\mathscr{A}$ scales like $N^{2\nu}$ [as suggested by (2.13)], we conclude that

$$\tau_{\text{int},\mathscr{A}}, \tau_{\text{int},|\mathscr{A}|} \geqslant \text{const} \times \langle N \rangle^{4\nu} \tag{A.50}$$

2. In *any* variable-N algorithm for SAWs that makes bounded changes in N at each move (e.g., the hybrid BFACF/cut-and-paste algorithm, which has $|\Delta N| \leqslant 2$), Theorem A.6 (and its corollary) and Theorem A.7 imply that

$$\tau_{\exp}, \tau_{\text{int},N} \geqslant \text{const} \times \langle N \rangle^2 \tag{A.51}$$

[assuming the usual scaling behavior (2.4) of the $c_N(x)$].

3. In the *fixed-N* local-deformation algorithms (see ref. 10 for references), we can use Theorem A.7 with f equal to the center of mass of the walk,

$$f(\omega) = \left| \frac{1}{N+1} \sum_{i=0}^{N} \omega_i \right| \tag{A.52}$$

Clearly f changes by at most order $1/N$ in a fixed-N local deformation.[25] On the other hand, the usual scaling behavior indicates that $\text{var}_\pi(f) \sim N^{2\nu}$. It follows from Theorem A.7 and (2.38) that

$$\tau_{\exp} \gtrsim \tau_{\text{int},f} \geqslant \text{const} \times N^{2+2\nu} \tag{A.53}$$

This lower bound is the rigorous version of the heuristic argument given in ref. 10.

ACKNOWLEDGMENTS

It is our pleasure to thank Neal Madras for many helpful discussions. We also thank an anonymous referee for several helpful suggestions. This research was supported in part by the Istituto Nazionale di Fisica Nucleare and by the U.S. National Science Foundation, grant DMS-8705599.

REFERENCES

1. C. Domb, *Adv. Chem. Phys.* **15**:229 (1969).
2. S. G. Whittington, *Adv. Chem. Phys.* **51**:1 (1982).
3. P. G. deGennes, *Phys. Lett. A* **38**:339 (1972).
4. J. des Cloizeaux, *J. Phys.* (Paris) **36**:281 (1975).
5. M. Daoud *et al.*, *Macromolecules* **8**:804 (1975).
6. V. J. Emery, *Phys. Rev. B* **11**:239 (1975).
7. C. Aragão de Carvalho, S. Caracciolo, and J. Fröhlich, *Nucl. Phys. B* **215**[FS7]:209 (1983).

[25] This is *not* the case for a $\Delta N = \pm 2$ BFACF move, which is why the argument does *not* apply in this case.

8. R. Fernández, J. Fröhlich, and A. D. Sokal, *Random Walks, Critical Phenomena, and Triviality in Quantum Field Theory* (Springer-Verlag, Berlin), to appear.
9. C. Aragão de Carvalho and S. Caracciolo, *J. Phys.* (Paris) **44**:323 (1983).
10. S. Caracciolo and A. D. Sokal, *J. Phys. A: Math. Gen.* **19**:L797 (1986).
11. S. Caracciolo and A. D. Sokal, *J. Phys. A: Math. Gen.* **20**:2569 (1987).
12. D. Stauffer and N. Jan, *Can. J. Phys.* **66**:187 (1988).
13. P. Devillard, *Physica A* **153**:189 (1988).
14. L. P. Kadanoff, *Physics* **2**:263 (1966).
15. M. E. Fisher, *Rep. Prog. Phys.* **30**:615 (1967).
16. C. K. Hall, *J. Stat. Phys.* **13**:157 (1975).
17. M. E. Fisher, in *Renormalization Group Methods in Statistical Physics and Quantum Field Theory*, M. S. Green and J. P. Gunton, eds. (Temple University Press, Philadelphia, 1973).
18. M. E. Fisher, in *Critical Phenomena* (Lecture Notes in Physics #186), F. J. W. Hahne, ed. (Springer-Verlag, Berlin, 1983).
19. H. J. F. Knops, J. M. J. van Leeuwen, and P. C. Hemmer, *J. Stat. Phys.* **17**:197 (1977).
20. M. E. Fisher and J.-H. Chen, *J. Phys.* (Paris) **46**:1645 (1985).
21. B. Berg and D. Foerster, *Phys. Lett.* **106B**:323 (1981).
22. A. D. Sokal and L. E. Thomas, *J. Stat. Phys.* **51**:907 (1988).
23. N. Madras and A. D. Sokal, *J. Stat. Phys.* **50**:109 (1988).
24. M. Lal, *Mol. Phys.* **17**:57 (1969).
25. B. MacDonald, N. Jan, D. L. Hunter, and M. O. Steinitz, *J. Phys. A: Math. Gen.* **18**:2627 (1985).
26. J. G. Curro, *J. Chem. Phys.* **61**:1203 (1974); **64**:2496 (1976).
27. H. L. Scott, Jr., *Biochem. Biophys. Acta* **469**:264 (1977).
28. L. E. Dubins, A. Orlitsky, J. A. Reeds, and L. A. Shepp, *IEEE Trans. Inf. Theory* **34**:1509 (1988).
29. N. Madras, A. Orlitsky, and L. A. Shepp, *J. Stat. Phys.* **58**:159 (1990).
30. E. J. Janse van Rensburg, S. G. Whittington, and N. Madras, The pivot algorithm and polygons: Results on the fcc lattice, *J. Phys. A: Math. Gen.* (to appear).
31. O. F. Olaj, W. Lantschbauer, and K. H. Pelinka, *Chemie-Kunststoffe Aktuell* **32**:199 (1978); O. F. Olaj and W. Lantschbauer, *Makromol. Chem. Rapid Commun.* **3**:847 (1982).
32. M. L. Mansfield, *J. Chem. Phys.* **77**:1554 (1982).
33. W. G. Madden, *J. Chem. Phys.* **87**:1405 (1987); **88**:3934 (1988).
34. J. Reiter, G. Zifferer, and O. F. Olaj, *Macromolecules* **22**:3120 (1989).
35. E. L. Pollock and D. M. Ceperley, *Phys. Rev. B* **30**:2555 (1984).
36. S. Caracciolo, A. Pelissetto, and A. D. Sokal, in *Lattice 88* (Proceedings of the 1988 Symposium on Lattice Field Theory, Fermilab, 22-25 September 1988), W. A. Bardeen *et al.*, eds. (North-Holland, Amsterdam, 1989) [= *Nucl. Phys. B (Proc. Suppl.)* **9**:525 (1989)].
37. J. M. Hammersley and K. W. Morton, *J. R. Stat. Soc. B* **16**:23 (1954).
38. J. M. Hammersley, *Proc. Camb. Phil. Soc.* **53**:642 (1957).
39. J. M. Hammersley and D. J. A. Welsh, *Q. J. Math.* (Oxford), *Ser. 2* **13**:108 (1962).
40. J. M. Hammersley, *Proc. Camb. Phil. Soc.* **57**:516 (1961).
41. H. Kesten, *J. Math. Phys.* **4**:960 (1963).
42. H. Kesten, *J. Math. Phys.* **5**:1128 (1964).
43. N. Madras, *J. Stat. Phys.* **53**:689 (1988).
44. N. Madras, Bounds on the critical exponent of self-avoiding polygons, York University, Department of Mathematics, Report no. 89-28 (1989).
45. G. Slade, *Commun. Math. Phys.* **110**:661 (1987).
46. G. Slade, *Ann. Prob.* **17**:91 (1989).

47. G. Slade, *J. Phys. A: Math. Gen.* **21**:L417 (1988).
48. H. E. Stanley, *Introduction to Phase Transitions and Critical Phenomena* (Oxford University Press, Oxford, 1971).
49. K. L. Chung, *Markov Chains with Stationary Transition Probabilities*, 2nd ed. (Springer, New York, 1967).
50. Z. Šidak, *Czechoslovak Math. J.* **14**:438 (1964).
51. A. D. Sokal and L. E. Thomas, *J. Stat. Phys.* **54**:797 (1989).
52. P. C. Hohenberg and B. I. Halperin, *Rev. Mod. Phys.* **49**:435 (1977).
53. J. G. Kemeny and J. L. Snell, *Finite Markov Chains* (Springer, New York, 1976).
54. M. Iosifescu, *Finite Markov Processes and Their Applications* (Wiley, Chichester, 1980).
55. A. J. Guttmann, *J. Phys. A: Math. Gen.* **20**:1839 (1987).
56. A. J. Guttmann, *J. Phys. A: Math. Gen.* **11**:L103 (1978).
57. N. Madras, unpublished (1986).
58. D. E. Knuth, *The Art of Computer Programming*, Vol. 3 (Addison-Wesley, Reading, Massachusetts, 1973), Section 6.4.
59. E. Horowitz and S. Sahni, *Fundamentals of Data Structures* (Computer Science Press, Potomac, Maryland, 1976), Section 9.3.
60. M. B. Priestley, *Spectral Analysis and Time Series* (Academic Press, London, 1981), Chapters 5–7.
61. R. H. Swendsen and J.-S. Wang, *Phys. Rev. Lett.* **58**:86 (1987).
62. X.-J. Li and A. D. Sokal, *Phys. Rev. Lett.* **63**:827 (1989).
63. M. Reed and B. Simon, *Functional Analysis* (Academic Press, New York, 1972).
64. T. Kato, *Perturbation Theory for Linear Operators*, 2nd ed. (Springer-Verlag, Berlin, 1976).
65. P. H. Peskun, *Biometrika* **60**:607 (1973).
66. K. Löwner, *Math. Z.* **38**:177 (1934).
67. W. F. Donoghue, Jr., *Monotone Matrix Functions and Analytic Continuation* (Springer-Verlag, New York, 1974).
68. R. Bellman, *Lin. Alg. Appl.* **1**:321 (1968).
69. W. N. Anderson, Jr., and R. J. Duffin, *J. Math. Anal. Appl.* **26**:576 (1969).
70. W. N. Anderson, Jr., and G. E. Trapp, *SIAM J. Appl. Math.* **28**:60 (1975).