Journal of Statistical Physics, Vol. 67, Nos. 1/2, 1992

Join-and-Cut Algorithm for Self-Avoiding Walks with Variable Length and Free Endpoints

Sergio Caracciolo,[1] Andrea Pelissetto,[2] and Alan D. Sokal[3]

Received July 18, 1991; final December 18, 1991

We introduce a new Monte Carlo algorithm for generating self-avoiding walks of variable length and free endpoints. The algorithm works in the unorthodox ensemble consisting of all pairs of SAWs such that the total number of steps $N_{\rm tot}$ in the two walks is fixed. The elementary moves of the algorithm are fixed-N (e.g., pivot) moves on the individual walks, and a novel "join-and-cut" move that concatenates the two walks and then cuts them at a random location. We analyze the dynamic critical behavior of the new algorithm, using a combination of rigorous, heuristic, and numerical methods. In two dimensions the autocorrelation time in CPU units grows as $N^{\approx 1.5}$, and the behavior improves in higher dimensions. This algorithm allows high-precision estimation of the critical exponent γ.

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

1. INTRODUCTION

Much progress has been made in the past 5–10 years in the development of new and more efficient Monte Carlo algorithms for simulating the self-avoiding walk (see Table I). In particular, the pivot algorithm[(1-3)],[4] is an extraordinarily efficient method for simulating SAWs of fixed length N and free endpoints: the autocorrelation time (for global observables, in CPU

[1] Scuola Normale Superiore and INFN-Sezione di Pisa, Pisa 56100, Italy.

[2] Department of Physics, Princeton University, Princeton, New Jersey 08544. Present address: Dipartimento di Fisica, Università degli Studi di Pisa, Pisa 56100, Italy.

[3] Department of Physics, New York University, New York, New York 10003.

[4] We take this opportunity to record yet another citation from the pre-1985 history of the pivot algorithm: Clark and Lal[(4)] employed the pivot algorithm to study self-avoiding lattice chains with an additional nearest-neighbor interaction energy. We thank Marvin Bishop for bringing this reference to our attention. Further references can be found in refs. 3 and 5.

Table I. Best Currently Available Algorithms for Monte Carlo Simulation of the Self-Avoiding Walk[a]

Ensemble	Quantities to be estimated	Algorithm	Autocorrelation time $\tau_{\mathrm{int}}^{(\mathrm{CPU})}$		
			$d=2$	$d=3$	$d=4$
Fixed N, free endpoints	ν	Pivot[1-3]	$\sim N$	$\sim N$	$\sim N$
Fixed N, fixed endpoints	ν	Cut-and-paste[6-8]	?	$\sim N^{\approx 1.1}$ (?)	?
Variable N, free endpoints	γ, μ, ν	Slithering tortoise[9-11] Incomplete enumeration[12]	$\sim\langle N\rangle^{\approx 2}$ $\sim\langle N\rangle^{\approx 2}$ (?)	$\sim\langle N\rangle^{\approx 2}$ $\sim\langle N\rangle^{\approx 2}$ (?)	$\sim\langle N\rangle^{2}$ $\sim\langle N\rangle^{2}$ (?)
Variable N, fixed endpoints	$\alpha_{\mathrm{sing}}, \mu, \nu$	BFACF[13-17,5] BFACF + cut-and-paste[5]	$\sim\langle N\rangle^{3}$ $\sim\langle N\rangle^{\approx 2.3}$	$\sim\langle N\rangle^{\approx 2.4}$ $\sim\langle N\rangle^{\approx 2.1}$ (?)	$\sim\langle N\rangle^{2}$ $\sim\langle N\rangle^{2}$ (?)
Two-SAW $\mathscr{T}_{N_{\mathrm{tot}}}$, free endpoints	γ, ν	Join-and-cut (this paper)	$\sim N_{\mathrm{tot}}^{\approx 1.5}$	$\sim N_{\mathrm{tot}}^{\approx 1.2}$ (?)	$\sim N_{\mathrm{tot}}^{\approx 1}$ (?)

[a] $\tau_{\mathrm{int}}^{(\mathrm{CPU})}$ denotes the integrated autocorrelation time of the algorithm, for global observables, in CPU time units.

units) grows only linearly with N.[3] Since it takes a time of order N merely to *write down* an N-step walk, the performance of the pivot algorithm is essentially optimal. Simulations in the fixed-length free-endpoint ensemble allow the high-precision estimation of the critical exponent ν, which governs the mean size of SAWs as a function of length.

Unfortunately, less progress has been made in simulating SAWs with *variable* lengths (and either free or fixed endpoints): the autocorrelation time of such algorithms grows at least as $\langle N\rangle^2$. Indeed, such a growth is inherent in *any* algorithm whose elementary moves make *bounded* changes in N: roughly speaking, N must perform a random walk on the nonnegative integers, and the autocorrelation time of such a random walk satisfies

$$\tau \gtrsim \mathrm{var}(N) \equiv \langle N^2\rangle - \langle N\rangle^2 \sim \langle N\rangle^2 \tag{1.1}$$

(ref. 5, Theorems A.6 and A.7). This is a shame, for the variable-length ensembles provide the only known efficient means for estimating the critical exponents γ and α_{sing} (and the connective constant μ), which govern the number of SAWs as a function of length.

In this paper[5] we propose an efficient—and amazingly simple—algorithm for simulating SAWs with variable lengths and *free* endpoints, and thereby estimating the critical exponent γ. In order to escape the bound (1.1), our algorithm makes *unbounded* (nonlocal) changes in N. To do this, it works in the unorthodox ensemble $\mathcal{T}_{N_{\mathrm{tot}}}$ consisting of all *pairs* of SAWs (ω_1, ω_2) (each walk starts at the origin and ends anywhere) such that the *total* number of steps in the two walks is some fixed number N_{tot}:

$$\begin{aligned}\mathcal{T}_{N_{\mathrm{tot}}} &\equiv \{(\omega_1, \omega_2)\colon \omega_1, \omega_2 \text{ are self-avoiding, with } |\omega_1| + |\omega_2| = N_{\mathrm{tot}}\}\\ &\equiv \bigcup_{N_1=0}^{N_{\mathrm{tot}}} \mathcal{S}_{N_1} \times \mathcal{S}_{N_{\mathrm{tot}}-N_1}\end{aligned} \tag{1.2}$$

Each pair (ω_1, ω_2) in this ensemble is given equal weight; therefore, the two walks are noninteracting except for the constraint on the sum of their lengths.

One sweep of the algorithm consists of two steps:

(a) We update independently each of the two walks, using some length-preserving ergodic algorithm. Here we will use the pivot algorithm.

(b) We perform a *join-and-cut* move: we concatenate the two walks ω_1 and ω_2, forming a new (not necessarily self-avoiding) walk

[5] A preliminary version of this work was reported at the Lattice '90 conference.[18]

$\omega_1 \circ \omega_2$; then we cut $\omega_1 \circ \omega_2$ at a random position, creating two new walks ω_1' and ω_2'. If ω_1' and ω_2' are both self-avoiding, we keep them; otherwise the move is rejected and we stay with ω_1 and ω_2.

The join-and-cut move is illustrated in Fig. 1.

Since the algorithm used in step (a) is ergodic in the ensemble of fixed-length walks, it is easy to see that the full algorithm is ergodic. (If ω_1 and ω_2 are perpendicular rods, then the join-and-cut move will always succeed.) It is also easy to see that the algorithm leaves invariant the equal-weight measure

$$\pi(\omega_1, \omega_2) = \frac{1}{Z(N_{\text{tot}})} \quad \text{for each} \quad (\omega_1, \omega_2) \in \mathscr{T}_{N_{\text{tot}}} \tag{1.3}$$

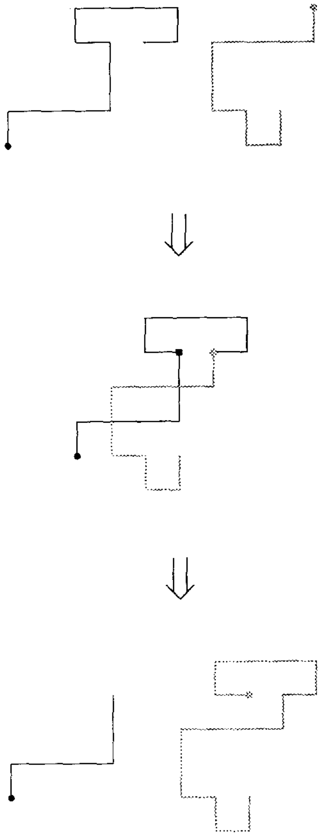

Fig. 1. Join-and-cut move: the two SAWs (upper figure) are concatenated (middle figure) and then cut at the point marked with a square (lower figure). Note that the concatenated walk *need not* be self-avoiding.

where

$$Z(N_{\text{tot}}) = \sum_{N_1=0}^{N_{\text{tot}}} c_{N_1} c_{N_{\text{tot}}-N_1} \tag{1.4}$$

Here c_N is the number of SAWs of length N, starting at the origin and ending anywhere; for large N it is believed to scale as

$$c_N \sim \mu^N N^{\gamma-1} \tag{1.5}$$

Therefore, from the probability distribution of the random variable $N_1 \equiv |\omega_1|$ (the length of the first walk) in the measure (1.3), we can obtain estimates of the critical exponent γ by the maximum-likelihood method.[9,19] Since the join-and-cut move can make large jumps in N_1 in a single step, this evades the bound (1.1).

The plan of this paper is as follows: In Section 2 we analyze the dynamic critical behavior of the algorithm, using a combination of rigorous mathematical arguments (Sections 2.1–2.3) and heuristic scaling arguments (Sections 2.4–2.5). In Section 3 we discuss the data structures needed in implementing the algorithm and analyze the computational complexity. In Section 4 we discuss the statistical issues that arise in the estimation of the critical exponent γ. In Section 5 we present some numerical results for two-dimensional self-avoiding walks. In Section 6 we summarize our results and discuss prospects for the future.

2. DYNAMIC CRITICAL BEHAVIOR

We want now to discuss the dynamic critical behavior of the algorithm defined in the preceding section. We shall make free use of the general theory of dynamic Monte Carlo methods (autocorrelation times τ_{exp} and $\tau_{\text{int},A}$, relation to spectral properties of the transition matrix, etc.); details of this theory can be found in refs. 3, 11, and 20.

2.1. Transition Matrices

The transition probability matrix P of the join-and-cut algorithm is a product of transition matrices P_a and P_b corresponding to steps (a) and (b) of the algorithm.

The form of P_a depends on the specific algorithm used in step (a). By hypothesis, P_a leaves invariant each subspace $\mathscr{S}_{N_1} \times \mathscr{S}_{N_2} \subset \mathscr{T}_{N_{\text{tot}}}$ and preserves the equal-weight distribution on this subspace; it therefore also preserves the equal-weight distribution π on $\mathscr{T}_{N_{\text{tot}}}$. (If, moreover, P_a satisfies detailed balance on each subspace $\mathscr{S}_{N_1} \times \mathscr{S}_{N_2}$, then it satisfies detailed balance on $\mathscr{T}_{N_{\text{tot}}}$. Detailed balance is convenient but not necessary.)

The transition matrix P_b is given by

$$P_b((\omega_1, \omega_2) \to (\omega_1', \omega_2')) = \begin{cases} \dfrac{1}{N_{\text{tot}}+1} & \text{if } \omega_1' \circ \omega_2' = \omega_1 \circ \omega_2 \text{ and } \\ & \quad \omega_1', \omega_2' \text{ are both self-avoiding} \\ & \quad \text{and } \omega_1' \neq \omega_1 \\ \dfrac{1 + C(\omega_1 \circ \omega_2)}{N_{\text{tot}}+1} & \text{if } \omega_1' = \omega_1 \text{ and } \omega_2' = \omega_2 \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

where $C(\omega_1 \circ \omega_2)$ is the number of splittings of $\omega_1 \circ \omega_2$ in which one of the pieces is *non*-self-avoiding. It is immediate that P_b is a symmetric matrix, i.e., it satisfies detailed balance with respect to the equal-weight distribution π. It also happens that P_b is positive-semidefinite. To see this, note first that P_b leaves invariant each subspace

$$\mathscr{T}^{\omega} = \{(\omega_1, \omega_2) \in \mathscr{T}_{N_{\text{tot}}} : \omega_1 \circ \omega_2 = \omega\} \tag{2.2}$$

where ω is a not-necessarily-self-avoiding walk of N_{tot} steps. The cardinality of $\mathscr{T}^{\omega}$ is $N_{\text{tot}} + 1 - C(\omega)$, and on this subspace P_b is given by

$$P_b \restriction \mathscr{T}^{\omega} = \frac{1}{N_{\text{tot}}+1} E + \frac{C(\omega)}{N_{\text{tot}}+1} I \tag{2.3}$$

where E is the matrix with all entries 1, and I is the identity matrix. Thus, the eigenvalues of $P_b \restriction \mathscr{T}^{\omega}$ are 1 (multiplicity 1) and $C(\omega)/(N_{\text{tot}}+1)$ [multiplicity $N_{\text{tot}} - C(\omega)$]. Since $C(\omega) \geqslant 0$, this completes the proof.

Remarks. 1. For a given pair $(\omega_1, \omega_2) \in \mathscr{S}_{N_1} \times \mathscr{S}_{N_2}$, a cutting point N_1' is allowed if and only if both of the subwalks $(\omega_1 \circ \omega_2)^{0, N_1'}$ and $(\omega_1 \circ \omega_2)^{N_1', N_{\text{tot}}}$ are self-avoiding. Therefore, the set of allowed cutting points N_1' is an interval $[N_{1,\min}(\omega_1 \circ \omega_2), N_{1,\max}(\omega_1 \circ \omega_2)]$, where $N_{1,\min}(\omega_1 \circ \omega_2) \leqslant N_1 \equiv |\omega_1| \leqslant N_{1,\max}(\omega_1 \circ \omega_2)$. It follows that

$$C(\omega_1 \circ \omega_2) = N_{\text{tot}} - [N_{1,\max}(\omega_1 \circ \omega_2) - N_{1,\min}(\omega_1 \circ \omega_2)] \tag{2.4}$$

2. A slight variant on this algorithm is to restrict the cutting point to lie in some specified interval $[N_{\min}^*, N_{\text{tot}} - N_{\min}^*]$, chosen with uniform

probability. This algorithm simulates the equal-weight ensemble on the configuration space

$$\mathscr{T}_{N_{\mathrm{tot}}, N^*_{\min}} \equiv \bigcup_{N_1 = N^*_{\min}}^{N_{\mathrm{tot}} - N^*_{\min}} \mathscr{S}_{N_1} \times \mathscr{S}_{N_{\mathrm{tot}} - N_1}$$

It is convenient to introduce also, for purposes of comparison, a *modified* join-and-cut move in which one insists that the concatenated walk $\omega_1 \circ \omega_2$ itself be self-avoiding:

(b′) We concatenate the two walks ω_1 and ω_2. If the concatenated walk $\omega_1 \circ \omega_2$ is self-avoiding, then we cut it at a random position, creating two new walks ω_1' and ω_2'. Otherwise the move is rejected and we stay with ω_1 and ω_2.

The corresponding transition matrix P_b' is given by

$$P_b'((\omega_1, \omega_2) \to (\omega_1', \omega_2')) \\ = \begin{cases} \dfrac{1}{N_{\mathrm{tot}} + 1} & \text{if } \ \omega_1' \circ \omega_2' = \omega_1 \circ \omega_2 \text{ and} \\ & \quad \ \omega_1 \circ \omega_2 \text{ is self-avoiding} \\ 1 & \text{if } \ \omega_1' = \omega_1 \text{ and } \omega_2' = \omega_2 \text{ and} \\ & \quad \ \omega_1 \circ \omega_2 \text{ is non-self-avoiding} \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

Clearly P_b' is symmetric. Its off-diagonal elements are less than or equal to those of P_b, so by a general comparison theorem (ref. 5, Theorem A.3) we have

$$P_b \leqslant P_b' \tag{2.6}$$

in quadratic-form sense. In particular, P_b' is positive-semidefinite. Indeed, P_b' leaves invariant each subspace $\mathscr{T}^\omega$, and on this subspace it is given by

$$P_b' \upharpoonright \mathscr{T}^\omega = \begin{cases} \dfrac{1}{N_{\mathrm{tot}} + 1} E & \text{if } \ \omega \text{ is self-avoiding} \\ I & \text{if } \ \omega \text{ is non-self-avoiding} \end{cases}$$

$$= \begin{cases} P_b \upharpoonright \mathscr{T}^\omega & \text{if } \ \omega \text{ is self-avoiding} \\ I & \text{if } \ \omega \text{ is non-self-avoiding} \end{cases} \tag{2.7}$$

We emphasize that the algorithm using rule (b′) is inferior to the algorithm using (b). This is obvious intuitively, since fewer useful moves are being made; and subject to certain conditions on P_a it will be proven rigorously at the end of Section 2.2. We therefore do *not* advocate using rule (b′) in practice. Its only role is to help simplify the analysis, as will be seen in Section 2.3.

2.2. Some Useful Abstract Nonsense

The transition matrices $P = P_a P_b$ and $P' = P_a P'_b$ are in general *nonsymmetric*, which somewhat complicates the mathematical analysis. *Suppose*, however, that P_a is symmetric (i.e., satisfies detailed balance) and positive-semidefinite (so that $0 \leqslant P_a \leqslant I$ in quadratic-form sense). Then it has a unique symmetric positive-semidefinite square root $P_a^{1/2}$ (which can be constructed, e.g., by diagonalizing P_a); this square root satisfies $0 \leqslant P_a^{1/2} \leqslant I$. The matrix $P_a^{1/2}$ leaves invariant each subspace $\mathscr{S}_{N_1} \times \mathscr{S}_{N_2}$ (because P_a does); and the constant function $\mathbf{1}$ is an eigenvector of $P_a^{1/2}$ with eigenvalue 1. It should be emphasized that the matrix $P_a^{1/2}$ is *not* in general a transition probability matrix, because its matrix elements may not all be nonnegative.[6] (In important special cases, $P_a^{1/2}$ may be a transition matrix: for example, if $P_a = Q_a^n$, where Q_a is a symmetric, positive-semidefinite transition matrix and n is *even*, then clearly $P_a^{1/2} = Q_a^{n/2}$ is a transition matrix.) Nevertheless, $P_a^{1/2}$ can be used in the mathematical analysis even if a direct probabilistic interpretation is lacking. To demonstrate this, we introduce the symmetric matrices $P_a^{1/2} P_b P_a^{1/2}$ and $P_a^{1/2} P'_b P_a^{1/2}$. We claim that:

(i) The eigenvalues of $P_a P_b$ are identical to those of $P_a^{1/2} P_b P_a^{1/2}$, counting multiplicity; in particular, they are real and lie in the interval $[0, 1]$.

(ii) For any observable $A(\omega_1, \omega_2)$ depending only on the *lengths* of the two walks [i.e., $A(\omega_1, \omega_2) = f(|\omega_1|)$], the autocorrelation function

$$C_{AA}(t) = (A, (P_a P_b)^t A) - (A, \mathbf{1})(\mathbf{1}, A) \tag{2.8}$$

[6] *Example*: Let

$$P_a = \begin{pmatrix} \alpha & 1/3 & 2/3-\alpha \\ 1/3 & 1/3 & 1/3 \\ 2/3-\alpha & 1/3 & \alpha \end{pmatrix}$$

with $1/3 \leqslant \alpha \leqslant 2/3$. Then

$$P_a^{1/2} = \begin{pmatrix} \alpha' & 1/3 & 2/3-\alpha' \\ 1/3 & 1/3 & 1/3 \\ 2/3-\alpha' & 1/3 & \alpha' \end{pmatrix}$$

with $\alpha' = 1/3 + [(3\alpha - 1)/6]^{1/2}$. For $\alpha > 5/9$ this implies $\alpha' > 2/3$, so that $P_a^{1/2}$ has some negative entries.

under the dynamics $P_a P_b$ is equal to the "autocorrelation function"

$$C_{AA}^{(\mathrm{sym})}(t) = (A, (P_a^{1/2} P_b P_a^{1/2})^t A) - (A, \mathbf{1})(\mathbf{1}, A) \qquad (2.9)$$

under the "dynamics" $P_a^{1/2} P_b P_a^{1/2}$.

Statement (i) follows from Lemma 2.1 below. It implies, among other things, that the exponential autocorrelation time τ_{exp} of the dynamics $P_a P_b$ can be determined by studying the eigenvalues of the symmetric matrix $P_a^{1/2} P_b P_a^{1/2}$.

Statement (ii) is an immediate consequence of the fact that P_a leaves invariant each subspace $\mathscr{S}_{N_1} \times \mathscr{S}_{N_2}$, so that $P_a A = P_a^{1/2} A = A$ for observables A depending only on N_1. It implies that for such observables we can learn everything about the dynamics—in particular, the integrated autocorrelation time $\tau_{\mathrm{int},A}$—by working with the symmetric matrix $P_a^{1/2} P_b P_a^{1/2}$.

Of course, analogous statements hold with P_b replaced by P'_b.

Lemma 2.1. Let P and Q be $n \times n$ complex matrices, with P Hermitian and positive-semidefinite. Then the matrices PQ and $P^{1/2} Q P^{1/2}$ have identical spectra, including multiplicity.

Proof. Let $\varepsilon > 0$; then the matrix $P + \varepsilon I$ is Hermitian and positive-definite, hence has a unique Hermitian positive-definite square root. Now $(P + \varepsilon I) Q$ is similar to $(P + \varepsilon I)^{1/2} Q (P + \varepsilon I)^{1/2}$, and so has identical spectrum (including multiplicity). Letting $\varepsilon \downarrow 0$, we have $(P + \varepsilon I) Q \to PQ$ and $(P + \varepsilon I)^{1/2} Q (P + \varepsilon I)^{1/2} \to P^{1/2} Q P^{1/2}$. Since the eigenvalues of a finite matrix are continuous functions of the matrix elements (this can be seen, e.g., from the characteristic polynomial), this proves the lemma. ▌

Remarks. 1. Although PQ and $P^{1/2} Q P^{1/2}$ have identical spectra, they need not have the same Jordan block structure, not even if Q is Hermitian: consider, for example,

$$P = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad Q = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

2. The analogue of Lemma 2.1 for bounded operators on a Hilbert space is much more subtle: the proof given here is valid only subject to restrictive hypotheses (e.g., P or Q is compact), and we do not know under what conditions the result is true. The trouble is that the spectrum is *not* in general a continuous function of the operator, even in the norm topology. For examples and discussion, see ref. 23, Problems 102–106. Nevertheless, we *can* prove that PQ and $P^{1/2} Q P^{1/2}$ have identical spectra (not necessarily with identical multiplicity) *except possibly for the point* 0.

This is a special case of the analogous result for operators AB and BA (ref. 23, Problem 76): just take $A = P^{1/2}$ and $B = P^{1/2}Q$. In particular, this shows that we can express the exponential autocorrelation time $\tau_{\text{exp}}(P_a P_b)$ for a Markov chain on an infinite state space in terms of the spectral radius of the self-adjoint operator $P_a^{1/2} P_b P_a^{1/2} \upharpoonright \mathbf{1}^{\perp}$.

Now we can prove that the algorithm using rule (b′) is inferior to the algorithm using (b), under the condition that P_a is symmetric and positive-semidefinite. Indeed, from $P_b \leqslant P'_b$ it follows that

$$P_a^{1/2} P_b P_a^{1/2} \leqslant P_a^{1/2} P'_b P_a^{1/2} \tag{2.10}$$

From this quadratic-form inequality, it follows by the min-max theorem[24] that the eigenvalues of $P_a^{1/2} P_b P_a^{1/2}$ are less than or equal to the corresponding eigenvalues of $P_a^{1/2} P'_b P_a^{1/2}$; combined with Lemma 2.1, this proves that

$$\tau_{\text{exp}}(P_a P_b) \leqslant \tau_{\text{exp}}(P_a P'_b) \tag{2.11}$$

On the other hand, (2.10) also implies that the integrated autocorrelation time for any observable A under the "dynamics" $P_a^{1/2} P_b P_a^{1/2}$ is less than or equal to that under $P_a^{1/2} P'_b P_a^{1/2}$ (ref. 5, Theorem A.2); combined with statement (ii) above, this proves that

$$\tau_{\text{int},A}(P_a P_b) \leqslant \tau_{\text{int},A}(P_a P'_b) \tag{2.12}$$

for all observables A depending only on the *lengths* of the walks.

2.3. Idealized Algorithm

In "hybrid" algorithms involving two radically different types of elementary moves—such as our steps (a) and (b)—physical insight can often be gained by studying first an *idealized* version of the algorithm in which one of the two moves is taken to be a "perfect" move of its type. (For previous examples of this method, see refs. 21 and 22.) Here we define an idealized version of the join-and-cut algorithm, for which we can prove a *rigorous upper bound* on the autocorrelation times τ_{exp} and $\tau_{\text{int},A}$. In this idealized algorithm, we use in step (a) a routine that produces a perfect random sample from the space of SAWs with given N_1, N_2: that is, the two new walks are *independent* of the previous pair and of each other. Mathematically, the idealized algorithm is defined by

$$P_a^{(\text{id})}((\omega_1, \omega_2) \to (\omega'_1, \omega'_2)) = \begin{cases} 1/c_{|\omega_1|} c_{|\omega_2|} & \text{if } \omega'_1 \in \mathcal{S}_{|\omega_1|} \text{ and } \omega'_2 \in \mathcal{S}_{|\omega_2|} \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

It should be noted that $P_a^{(\mathrm{id})}$ is symmetric and idempotent, i.e., $P_a^{(\mathrm{id})^2} = P_a^{(\mathrm{id})} = P_a^{(\mathrm{id})^T}$. In particular, it is positive-semidefinite, with eigenvalues 0 and 1; and we have $P_a^{(\mathrm{id})^{1/2}} = P_a^{(\mathrm{id})}$. Lemma 2.1 then implies that the eigenvalues of the (nonsymmetric) matrix $P^{(\mathrm{id})} = P_a^{(\mathrm{id})} P_b$ are the same as those of the symmetric matrix $P_a^{(\mathrm{id})} P_b P_a^{(\mathrm{id})}$.

In our case P_b is symmetric and positive-semidefinite, hence so is $P_a^{(\mathrm{id})} P_b P_a^{(\mathrm{id})}$. It follows that the eigenvalues of $P_a^{(\mathrm{id})} P_b P_a^{(\mathrm{id})}$, call them $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$ $(n = |\mathscr{T}_{N_{\mathrm{tot}}}|)$, lie in the interval $[0, 1]$. The constant function $\mathbf{1}$ is an eigenvector corresponding to the eigenvalue $\lambda_1 = 1$, and this eigenvalue is simple (because the algorithm is ergodic). The exponential autocorrelation time τ_{exp} of the idealized algorithm $P^{(\mathrm{id})} = P_a^{(\mathrm{id})} P_b$ or its symmetrization $P_a^{(\mathrm{id})} P_b P_a^{(\mathrm{id})}$ is given by

$$\exp(-1/\tau_{\mathrm{exp}}) = \lambda_2(P_a^{(\mathrm{id})} P_b) = \lambda_2(P_a^{(\mathrm{id})} P_b P_a^{(\mathrm{id})}) \tag{2.14}$$

So we need to control the eigenvalues of the symmetric matrix $P_a^{(\mathrm{id})} P_b P_a^{(\mathrm{id})}$.

Let us remark that a completely analogous analysis holds for the *modified* idealized algorithm $P'^{(\mathrm{id})} = P_a^{(\mathrm{id})} P'_b$ and its symmetrization $P_a^{(\mathrm{id})} P'_b P_a^{(\mathrm{id})}$. From (2.6) we deduce

$$P_a^{(\mathrm{id})} P_b P_a^{(\mathrm{id})} \leqslant P_a^{(\mathrm{id})} P'_b P_a^{(\mathrm{id})} \tag{2.15}$$

in quadratic-form sense. It follows from the min-max theorem[24] that each eigenvalue of $P_a^{(\mathrm{id})} P_b P_a^{(\mathrm{id})}$ is less than or equal to the corresponding eigenvalue of $P_a^{(\mathrm{id})} P'_b P_a^{(\mathrm{id})}$:

$$\lambda_k(P_a^{(\mathrm{id})} P_b P_a^{(\mathrm{id})}) \leqslant \lambda_k(P_a^{(\mathrm{id})} P'_b P_a^{(\mathrm{id})}) \tag{2.16}$$

for $1 \leqslant k \leqslant n$.

Since $P_a^{(\mathrm{id})}$ provides perfect randomization within each subspace $\mathscr{S}_{N_1} \times \mathscr{S}_{N_2}$, the dynamics of the Markov chain $P_a^{(\mathrm{id})} P_b P_a^{(\mathrm{id})}$ is determined completely by the behavior of the corresponding *aggregated Markov chain*[25,11] with state space $\{0, 1,..., N_{\mathrm{tot}}\}$, defined by

$$\bar{P}(N_1 \to N'_1) = \frac{1}{c_{N_1} c_{N_{\mathrm{tot}} - N_1}} \sum_{\substack{(\omega_1, \omega_2) \in \\ \mathscr{S}_{N_1} \times \mathscr{S}_{N_{\mathrm{tot}} - N_1}}} \sum_{\substack{(\omega'_1, \omega'_2) \in \\ \mathscr{S}_{N'_1} \times \mathscr{S}_{N_{\mathrm{tot}} - N'_1}}} P_b((\omega_1, \omega_2) \to (\omega'_1, \omega'_2)) \tag{2.17}$$

(More precisely, the nonzero eigenvalues of $P_a^{(\mathrm{id})} P_b P_a^{(\mathrm{id})}$ are exactly those of $\bar{P}$.) Inserting (2.1), we find that the off-diagonal matrix elements of $\bar{P}$ are given by

$$\bar{P}(N_1 \to N'_1) = \frac{F(N_1, N'_1; N_{\mathrm{tot}})}{(N_{\mathrm{tot}} + 1)\, c_{N_1} c_{N_{\mathrm{tot}} - N_1}} \quad \text{for} \quad N_1 \neq N'_1 \tag{2.18}$$

where $F(N_1, N_1'; N_{\text{tot}})$ is the number of not-necessarily-self-avoiding walks ω of N_{tot} steps such that all of the subwalks ω^{0,N_1}, $\omega^{N_1,N_{\text{tot}}}$, $\omega^{0,N_1'}$, and $\omega^{N_1',N_{\text{tot}}}$ are self-avoiding. Since $F(N_1, N_1'; N_{\text{tot}})$ is obviously symmetric under $N_1 \leftrightarrow N_1'$, we see that $\bar{P}$ satisfies detailed balance with respect to the probability distribution

$$\bar{\pi}(N_1) = \frac{c_{N_1} c_{N_{\text{tot}} - N_1}}{Z(N_{\text{tot}})} \tag{2.19}$$

which is the aggregated probability distribution corresponding to π. Unfortunately, we know very little else about $F(N_1, N_1'; N_{\text{tot}})$; but it is in any case larger than or equal to $c_{N_{\text{tot}}}$, the number of *self-avoiding* walks of N_{tot} steps. So let us consider the *modified* aggregated Markov chain $\bar{P}'$ defined by

$$\bar{P}'(N_1 \to N_1') = \frac{c_{N_{\text{tot}}}}{(N_{\text{tot}} + 1)\, c_{N_1} c_{N_{\text{tot}} - N_1}} \qquad \text{for} \quad N_1 \neq N_1' \tag{2.20}$$

(This is the aggregated Markov chain corresponding to the modified transition matrix $P_a^{(\text{id})} P_b' P_a^{(\text{id})}$.) The matrices $\bar{P}$ and $\bar{P}'$ both satisfy detailed balance for the same distribution $\bar{\pi}$, and the off-diagonal elements of $\bar{P}'$ are smaller than or equal to those of $\bar{P}$. Hence $\bar{P} \leqslant \bar{P}'$ in the sense of quadratic forms, and each eigenvalue of $\bar{P}$ is less than or equal to the corresponding eigenvalue of $\bar{P}'$. In particular, $\lambda_2(\bar{P}) \leqslant \lambda_2(\bar{P}')$.

So it suffices to study the eigenvalues of the matrix $\bar{P}'$. Now

$$\bar{P}'_{ij} = (1 - nd_i)\, \delta_{ij} + d_i \tag{2.21}$$

where $n \equiv N_{\text{tot}} + 1$ and $d_i \equiv c_{N_{\text{tot}}}/(nc_i c_{N_{\text{tot}} - i})$. Let D be the diagonal matrix $D_{ij} = d_i \delta_{ij}$; then

$$\begin{aligned}(D^{-1/2} \bar{P}' D^{1/2})_{ij} &= (1 - nd_i)\, \delta_{ij} + d_i^{1/2} d_j^{1/2} \\ &\equiv R_{ij} + S_{ij}\end{aligned} \tag{2.22}$$

Now R is a symmetric matrix with eigenvalues $1 - nd_i$ $(i = 0, ..., N_{\text{tot}})$, and S is a symmetric positive-semidefinite matrix of rank 1. By Weyl's interlacing theorem (ref. 26, Theorem 3.6.3), the eigenvalues of the symmetric matrix $D^{-1/2} \bar{P}' D^{1/2}$ interlace those of R, so that

$$\lambda_k(R) \leqslant \lambda_k(\bar{P}') = \lambda_k(D^{-1/2} \bar{P}' D^{1/2}) \leqslant \lambda_{k-1}(R) \tag{2.23}$$

for $2 \leqslant k \leqslant n$. In particular,

$$\max_{0 \leqslant i \leqslant N_{\text{tot}}}{}' \; (1 - nd_i) \leqslant \lambda_2(\bar{P}') \leqslant \max_{0 \leqslant i \leqslant N_{\text{tot}}} (1 - nd_i) \tag{2.24}$$

where max′ denotes second-largest. Now it follows from the assumed behavior (1.5) that[7] $c_i c_{N_{\mathrm{tot}}-i}$ is maximum at or near $i = N_{\mathrm{tot}}/2$, and that

$$\Big(\max_{0 \leqslant i \leqslant N_{\mathrm{tot}}}{}' \; c_i c_{N_{\mathrm{tot}}-i}\Big)/c_{N_{\mathrm{tot}}}, \; \Big(\max_{0 \leqslant i \leqslant N_{\mathrm{tot}}} c_i c_{N_{\mathrm{tot}}-i}\Big)/c_{N_{\mathrm{tot}}} \sim N_{\mathrm{tot}}^{\gamma-1} \tag{2.25}$$

Hence

$$\lambda_2(\bar{P}) \leqslant \lambda_2(\bar{P}') \approx 1 - \frac{\mathrm{const}}{N^{\gamma-1}} \tag{2.26}$$

In summary, we have proven [subject to the assumption (1.5)] that

$$\tau_{\mathrm{exp}} \sim N^{\gamma-1} \quad \text{for the \textit{modified} idealized algorithm} \tag{2.27a}$$

$$\tau_{\mathrm{exp}} \lesssim N^{\gamma-1} \quad \text{for the original idealized algorithm} \tag{2.27b}$$

(We expect that $\tau_{\mathrm{exp}} \sim N^{\gamma-1}$ also for the original idealized algorithm, but we do not have a rigorous proof.) The general theory of reversible Markov chains[3,20,5] then implies that

$$\tau_{\mathrm{int},A} \lesssim N^{\gamma-1} \tag{2.28}$$

for any observable A, for both the modified and original idealized algorithms. [We expect that $\tau_{\mathrm{int},A} \sim N^{\gamma-1}$ for "reasonable" observables A, such as functions $f(N_1)$ that are even under the symmetry $N_1 \to N_{\mathrm{tot}} - N_1$; but we do not have a rigorous proof.]

Remark. The quantity $f_i \equiv nd_i$, which appears in (2.24), is the mean acceptance fraction of modified join-and-cut moves starting from $N_1 = i$. The mean acceptance fraction, averaging over all N_1, is

$$f \equiv \sum_{i=0}^{N_{\mathrm{tot}}} \frac{c_i c_{N_{\mathrm{tot}}-i}}{Z(N_{\mathrm{tot}})} f_i = \frac{(N_{\mathrm{tot}}+1)\, c_{N_{\mathrm{tot}}}}{Z(N_{\mathrm{tot}})} \tag{2.29}$$

It follows from (1.5) that

$$f_i \sim N^{-(\gamma-1)} \qquad \text{if} \quad i, N_{\mathrm{tot}} - i \sim N_{\mathrm{tot}} \tag{2.30a}$$

$$f \sim N^{-(\gamma-1)} \tag{2.30b}$$

The upshot of the foregoing proof is that one successful *idealized* join-and-cut move is sufficient to generate an "effectively independent" configuration (ω_1', ω_2')—hence $\tau_{\mathrm{exp}} \sim 1/f$.

[7] It is an interesting and (as far as we know) open problem to prove *rigorously* that $c_i c_{N_{\mathrm{tot}}-i}$ takes its maximum at $i = [N_{\mathrm{tot}}/2]$. Indeed, proving this for each even N_{tot} is equivalent to proving that $\log c_N$ is a *concave* function of N—an assertion which appears to be true but very difficult to prove. We thank Neal Madras for a discussion of this point.

2.4. Review of Pivot Algorithm

In the *practical* version of the join-and-cut algorithm, we will use pivot moves for step (a). We begin, therefore, by reviewing the known facts[3] on the behavior of the pivot algorithm.

The pivot algorithm for N-step SAWs (starting at the origin and ending anywhere) is defined as follows: Choose randomly, with uniform probability, an integer $k \in \{0, 1,..., N-1\}$.[8] Choose randomly an element g of the lattice symmetry group (i.e., lattice rotations and reflections).[9] Propose a pivot move $\omega \rightarrow \omega'$ defined by

$$\omega_i' = \begin{cases} \omega_i & \text{for} \quad 0 \leqslant i \leqslant k \\ \omega_k + g(\omega_i - \omega_k) & \text{for} \quad k+1 \leqslant i \leqslant N \end{cases} \tag{2.31}$$

where i labels sites along the walk. The proposed move is accepted if ω' is self-avoiding; otherwise, it is rejected and we stay with ω.

Remark. For certain (but not all) choices of the transition probabilities Prob(g), the transition matrix P_a of the pivot algorithm is positive-semidefinite. This occurs, for example, if Prob(g) is uniform on the lattice symmetry group (*including* the identity element); the proof is similar to the one at (2.1)–(2.3).

In ref. 3 it is argued heuristically, and confirmed numerically, that the acceptance fraction f(pivot) and autocorrelation times $\tau_{\mathrm{int},A}$(pivot) and τ_{exp}(pivot) behave as

$$f(\text{pivot}) \sim N^{-p} \tag{2.32}$$

$$\tau_{\mathrm{int},A}(\text{pivot}) \sim \begin{cases} N^p & \text{for } \textit{global} \text{ observables } A \\ N^{1+p} & \text{for } \textit{local} \text{ observables } A \end{cases} \tag{2.33}$$

$$\tau_{\mathrm{exp}}(\text{pivot}) \sim N^{1+p} \tag{2.34}$$

where $p \approx 0.19$ in dimension $d = 2$,[3] and $p \approx 0.11$ in $d = 3$.[27] Here a "global" observable is one that receives contributions equally from all steps of the walk; examples are the end-to-end distance and the radius of gyration. A "local" observable is one that depends primarily on a very small part of the walk; an example is the angle between the 17th and 18th

[8] In one variant of the pivot algorithm (ref. 3, p. 117), the starting point of the walk ($k = 0$) is excluded as a possible pivot point, since pivoting there corresponds simply to an overall rotation or reflection of the walk. This variant is fine for studying a single walk, but it is *not* appropriate for applications involving *pairs* of walks, as is the case here.

[9] The probability distribution must satisfy Prob(g) = Prob(g^{-1}) in order to satisfy detailed balance; and "enough" of the probabilities must be nonzero so as to make the algorithm ergodic. See refs. 3 and 7 for details.

steps of the walk. The heuristic reasoning behind (2.33) is the following: The pivot moves are very radical; after a few successful pivots, the *global* conformation of the walk should have reached an "essentially new" state. So we expect $\tau_{\text{int},A}(\text{pivot}) \sim 1/f(\text{pivot}) \sim N^p$ for *global* observables A. On the other hand, for a *local* observable to change, it is necessary to have a successful pivot *in a small portion of the walk*; this requires a time $\sim N/f(\text{pivot}) \sim N^{1+p}$. More generally, for observables A that depend primarily on $\approx M$ steps of the walk ($M \leqslant N$), we expect $\tau_{\text{int},A}(\text{pivot}) \sim N^{1+p}/M$.

2.5. Practical Algorithm

Now we consider the *practical join-and-cut algorithm* in which step (a) consists of n_{piv} pivot-algorithm attempts on each of the walks ω_1, ω_2. Our treatment will be heuristic rather than rigorous.

It is reasonable to conjecture that the autocorrelation time for any observable A follows a scaling law

$$\tau_{\text{int},A} \sim N_{\text{tot}}^{r_A} F_A(n_{\text{piv}}/N_{\text{tot}}^{s_A}) \tag{2.35}$$

as $N_{\text{tot}} \to \infty$, for some exponents r_A, s_A and some scaling function F_A. Our greatest interest is in observables A that are functions only of the *lengths* of the two walks and are invariant under $N_1 \leftrightarrow N_2$: that is, $A = f(N_1)$ for some f satisfying $f(N_1) = f(N_{\text{tot}} - N_1)$.

To extract the exponents r_A, s_A and the asymptotic properties of the scaling function F_A, we consider three ways of taking N_{tot} to infinity:

1. First take $n_{\text{piv}} \to \infty$, *then* take $N_{\text{tot}} \to \infty$.
2. Take $N_{\text{tot}} \to \infty$ with $n_{\text{piv}} \sim N_{\text{tot}}^{s_A}$.
3. Take $N_{\text{tot}} \to \infty$ with n_{piv} fixed.

1. $n_{\text{piv}} \to \infty$ *followed by* $N_{\text{tot}} \to \infty$. As $n_{\text{piv}} \to \infty$, the practical algorithm approaches the idealized algorithm. Indeed, for $n_{\text{piv}} \gg \tau_{\text{exp}}(\text{pivot}, N \sim N_{\text{tot}}) \sim N_{\text{tot}}^{1+p}$, the autocorrelation time of the practical algorithm should be essentially independent of n_{piv} (with corrections expected to be exponentially small in $n_{\text{piv}}/N_{\text{tot}}^{1+p}$). Combining this fact with the scaling Ansatz (2.35) and the conjectured behavior $\tau_{\text{int},A}^{(\text{id})} \sim N^{\gamma-1}$, we conclude that

$$r_A = \gamma - 1 \tag{2.36}$$

$$s_A \leqslant 1 + p \tag{2.37}$$

and

$$0 < F_A(\infty) \equiv \lim_{z \to \infty} F_A(z) < \infty \tag{2.38}$$

Waving our hands a little bit more, we can argue that $F_A(z)$ should approach its limiting value $F_A(\infty)$ at the rate

$$F_A(z) - F_A(\infty) \sim \frac{1}{z^{\approx 1}} \qquad \text{as} \quad z \to \infty \tag{2.39}$$

Indeed, in the pivot algorithm for an *ordinary* random walk, the autocorrelation functions for *global* observables decay like $1/t$ in the range $1 \ll t \ll \tau_{\text{exp}}(\text{pivot}) \sim N$ (ref. 3, Section 3.3); and we expect a similar behavior for a self-avoiding walk, i.e., a decay like $1/t^{\approx 1}$ for $N^p \ll t \ll \tau_{\text{exp}}(\text{pivot}) \sim N^{1+p}$. Assuming that the behavior of the join-and-cut moves depends on a *global* observable (see below for further discussion), this suggests that the deviations from the idealized join-and-cut algorithm should scale like $1/n_{\text{piv}}^{\approx 1}$ for $N_{\text{tot}}^p \ll n_{\text{piv}} \ll N_{\text{tot}}^{1+p}$. Provided that $s_A > p$ [as we argue below, see (2.43)], this implies (2.39).

In Section 5.1 we shall test numerically the predictions (2.36)–(2.39).

2. $n_{\text{piv}} \sim N_{\text{tot}}^{s_A}$. If $z \equiv n_{\text{piv}}/N_{\text{tot}}^{s_A}$ is fixed ($0 < z < \infty$), then we have

$$\frac{\tau_{\text{int},A}}{\tau_{\text{int},A}^{(\text{id})}} = \frac{F_A(z)}{F_A(\infty)} = \text{some finite number} \tag{2.40}$$

Thus, determining the exponent s_A amounts to answering the question: Roughly how many pivot moves do we have to perform so that the practical algorithm will be only a finite factor worse than the idealized algorithm? It is tempting to guess that the answer is: a few pivot-algorithm autocorrelation times. But this raises the question: *which* autocorrelation time? Is it $\tau_{\text{int},B}(\text{pivot})$, and if so, for which observable $B(\omega_1, \omega_2)$? One might guess that the relevant observables B are those that are somehow related to the behavior of the join-and-cut move, i.e., which are functions of $N_{1,\min}(\omega_1 \circ \omega_2)$ and $N_{1,\max}(\omega_1 \circ \omega_2)$. For example, one might consider

$$\begin{aligned} J(\omega_1, \omega_2) &\equiv \text{Prob(a join-and-cut move on } \omega_1, \omega_2 \text{ will be successful)} \\ &= \frac{N_{1,\max}(\omega_1 \circ \omega_2) - N_{1,\min}(\omega_1 \circ \omega_2) + 1}{N_{\text{tot}} + 1} \end{aligned} \tag{2.41}$$

or

$$\begin{aligned} \mathscr{A}_f(\omega_1, \omega_2) &\equiv \langle f(|\omega_1'|) \rangle_{\omega_1, \omega_2} \\ &= \frac{1}{N_{\text{tot}} + 1} \sum_{N_1' = N_{1,\min}(\omega_1 \circ \omega_2)}^{N_{1,\max}(\omega_1 \circ \omega_2)} f(N_1') \; + \; [1 - J(\omega_1, \omega_2)] \, f(|\omega_1|) \end{aligned} \tag{2.42}$$

and similar observables. (Here $\langle \cdot \rangle_{\omega_1, \omega_2}$ denotes a conditional expectation given ω_1, ω_2.) It is far from obvious to us how $\tau_{\mathrm{int}, B}(\mathrm{pivot})$ for these observables scales. These observables seem to be somewhere between "global" and "local": they depend on the entire walks ω_1 and ω_2, but depend most strongly on the parts of ω_1 and ω_2 near the joining point.[10] So we would guess that

$$p < s_A < 1 + p \tag{2.43}$$

Unfortunately this does not constrain s_A terribly closely.

Waving our hands a little bit more, we can argue that these observables depend primarily on the parts of ω_1 and ω_2 in an interval of width $\approx M \sim N_{\mathrm{tot}}^{2-\gamma}$ near the joining point: the idea is that if a join-and-cut move is to *fail*, then this failure will be detected at an average distance of order $N_{\mathrm{tot}}^{2-\gamma}$ away from the joining point (see Section 3.1 below). Then the heuristic argument sketched at the end of Section 2.4 suggests that $\tau_{\mathrm{int}, B}(\mathrm{pivot}) \sim N_{\mathrm{tot}}^{1+p}/M \sim N_{\mathrm{tot}}^{p+\gamma-1}$. Hence we predict (though not with great confidence) that

$$s_A = p + \gamma - 1 \tag{2.44}$$

In Section 5.1 we shall estimate numerically s_A (and the corresponding scaling function F_A) by fitting to the scaling form (2.35).

3. n_{piv} *fixed.* If

$$F_A(z) \sim z^{-a_A} \qquad \text{as} \quad z \to 0 \tag{2.45}$$

then

$$\tau_{\mathrm{int}, A} \sim N_{\mathrm{tot}}^{q_A} \equiv N_{\mathrm{tot}}^{r_A + a_A s_A} \tag{2.46}$$

as $N_{\mathrm{tot}} \to \infty$ with n_{piv} fixed (>0). In Section 5.1 we shall estimate the exponent $q_A \equiv r_A + a_A s_A$ by fitting to (2.46), and shall also estimate a_A by comparing the estimated scaling function F_A to (2.45).

Remark. It is worth remarking that the autocorrelation function *at time* 1 of any observable $f(N_1)$ is the same in the practical algorithm as

[10] Indeed, if $\omega_1, \tilde{\omega}_1 \in \mathscr{S}_{N_1}$ and $\omega_2, \tilde{\omega}_2 \in \mathscr{S}_{N_2}$ are such that the subwalks $(\omega_1 \circ \omega_2)^{k, N_1}$ and $(\tilde{\omega}_1 \circ \tilde{\omega}_2)^{k, N_1}$ are equal and *non*-self-avoiding (for some $k \leqslant N_1$), then $N_{1,\mathrm{min}}(\omega_1 \circ \omega_2) = N_{1,\mathrm{min}}(\tilde{\omega}_1 \circ \tilde{\omega}_2) \geqslant k$ irrespective of the behavior of the first k steps of ω_1 and $\tilde{\omega}_1$. An analogous statement holds true regarding the subwalks $[N_1, l]$ and the resulting values of $N_{1,\mathrm{max}}$.

in the idealized algorithm. Indeed, for any observable f depending only on N_1, we have $P_a f = f$, and hence

$$\begin{aligned} C_{ff}(1) &= (f, P_a P_b f) - (f, 1)(1, f) \\ &= (f, P_b f) - (f, 1)(1, f) \end{aligned} \tag{2.47}$$

which is independent of the particular choice of updating P_a. Of course, this identity does *not* persist at times $t \geqslant 2$, because $P_b P_a f = P_b f$ is not a function of N_1 alone.

3. DATA STRUCTURES AND COMPUTATIONAL COMPLEXITY

In this section we discuss the data structures needed for implementing the pivot and join-and-cut moves, and the computational complexity of the resulting algorithm.

3.1. Basic Method

The implementation of the pivot algorithm is discussed in ref. 3, Section 3.4; let us review it briefly. For checking self-avoidance we use a *hash table* with linear probing[28]; each hash-table query, insertion or deletion takes a mean time of order 1. To test whether a proposed new walk ω' is self-avoiding, we insert the proposed points ω'_i into an initially empty hash table, checking for intersections. Since the most probable intersections are those *near* the pivot point, we insert the points ω'_i into the hash table *starting from the pivot point and working outward* (stopping as soon as an intersection is detected): the mean work for a *failed* pivot move is then argued to scale as

$$E(\text{work} \mid \text{failure}) = N^{1-p} \tag{3.1}$$

where p is the acceptance-fraction exponent defined in (2.32). The work for a successful pivot move is of course of order N. The mean work per pivot move is thus

$$\begin{aligned} E(\text{work}) &\sim \text{Prob(success)}\, E(\text{work} \mid \text{success}) + \text{Prob(failure)}\, E(\text{work} \mid \text{failure}) \\ &\sim N^{-p} \cdot N + 1 \cdot N^{1-p} \\ &\sim N^{1-p} \end{aligned} \tag{3.2}$$

This is significantly smaller than the behavior $E(\text{work}) \sim N$ of the naive algorithm in which the points ω'_i are inserted in no special order. Finally, the hash table is cleared after each use; to facilitate this, we keep a linear list of the occupied hash-table entries. Obviously the work for clearing the hash table scales in the same way as the work for creating it.

Next we consider the join-and-cut moves. Suppose first that we are given the walks ω_1 and ω_2 in the form of linear lists, and that we are given the proposed cutting point N_1'. How do we test ω_1' and ω_2' for self-avoidance? Note first that one of the two walks will *decrease* in length, and this walk will automatically be self-avoiding. Therefore we need only check self-avoidance of the other walk (the one that *grows*); this is done exactly as in the pivot algorithm, *working outward from the joining point* (and of course stopping as soon as an intersection is detected). The work required for this computation is therefore proportional to the distance from the joining point to the *first* intersection. Let us estimate the mean work.

We must attach a subwalk of length $M \equiv |N_1' - N_1|$ to a walk of length $N \equiv N_1$ or N_2. (Typically M and N are both of order N_{tot}.) This subwalk is, by definition, the first (or last) M steps of a SAW of length $N_{\text{tot}} - N$; but it is a reasonable *approximation* to pretend that it is simply a random M-step SAW.[11] [This same approximation was used in ref. 3, Section 3.4 in deriving (3.1).] With this approximation, the acceptance fraction (for fixed M, N) is

$$f_{M,N} \approx \frac{c_{N+M}}{c_N c_M} \tag{3.3}$$

where c_N is the number of SAWs of length N. From the scaling Ansatz (1.5), we find

$$f_{M,N} \sim \min(M, N)^{-(\gamma - 1)} \tag{3.4}$$

To find out *where* an intersection first occurs, we apply this same approximation also to *subwalks* of the two walks. That is, the probability that i steps of each of the walks can be joined *without* finding an intersection is approximately

$$f_{i,i} \approx \frac{c_{2i}}{c_i^2} \sim i^{-(\gamma - 1)} \tag{3.5}$$

for $1 \leqslant i \leqslant \min(M, N)$, and

$$f_{\min(M,N),i} \approx \frac{c_{\min(M,N)+i}}{c_{\min(M,N)} c_i} \sim \min(M, N)^{-(\gamma - 1)} \tag{3.6}$$

for $\min(M, N) < i \leqslant \max(M, N)$.

[11] Note that the probability distribution of the first M steps of a SAW of length $N_{\text{tot}} - N$ is *not* the same as the equal-weight distribution on M-step SAWs. In the former distribution, each M-step SAW gets weight proportional to the number of ways it can be extended to an $(N_{\text{tot}} - N)$-step SAW; and these weights are *not* equal. (Indeed, some M-step SAWs have *zero* weight, such as those ending in culs-de-sac.)

Let us now define the "work" to be k if an intersection is found at step k, and to be $\max(M, N)+1$ if no intersection is found at all (i.e., if the join-and-cut move is successful).[12] That is,

$$\begin{aligned}\text{Prob}(\text{work} > k) &= \text{Prob}(\text{no intersection found up to step } k)\\ &\approx \begin{cases} f_{k,k} & \text{for} \quad 1 \leqslant k \leqslant \min(M, N) \\ f_{\min(M,N),k} & \text{for} \quad \min(M, N) < k \leqslant \max(M, N) \end{cases}\end{aligned} \tag{3.7}$$

Then the mean work expended in *failed* join-and-cut attempts is

$$\begin{aligned}&E(\text{work in failures})\\ &\quad = \sum_{j=1}^{\max(M,N)} j \,\text{Prob}(\text{work} = j)\\ &\quad = \sum_{k=0}^{\max(M,N)} \text{Prob}(\text{work} > k)\\ &\quad \sim \sum_{k=1}^{\min(M,N)} k^{-(\gamma-1)} + \sum_{k=\min(M,N)+1}^{\max(M,N)} \min(M, N)^{-(\gamma-1)}\\ &\quad \sim \min(M, N)^{2-\gamma} + [\max(M, N) - \min(M, N)] \min(M, N)^{1-\gamma}\\ &\quad \sim N_{\text{tot}}^{2-\gamma}\end{aligned} \tag{3.8}$$

(Here we take for granted that $\gamma < 2$.) Since the probability of failure is $O(1)$ as $N_{\text{tot}} \to \infty$, the mean work *conditional on failure* is

$$E(\text{work} \mid \text{failure}) \sim N_{\text{tot}}^{2-\gamma} \tag{3.9}$$

On the other hand, a successful join-and-cut move requires a work $\max(M, N)+1$, so

$$E(\text{work} \mid \text{success}) \sim N_{\text{tot}} \tag{3.10}$$

Thus, the mean work per join-and-cut move is

$$\begin{aligned}E(\text{work}) &\sim \text{Prob}(\text{success})\, E(\text{work} \mid \text{success}) + \text{Prob}(\text{failure})\, E(\text{work} \mid \text{failure})\\ &\sim N_{\text{tot}}^{-(\gamma-1)} \cdot N_{\text{tot}} + 1 \cdot N_{\text{tot}}^{2-\gamma}\\ &\sim N_{\text{tot}}^{2-\gamma}\end{aligned} \tag{3.11}$$

[12] This definition is taken for simplicity; it differs from the number of hash-table insertions by a factor between 1 and 2. Such a bounded factor is of course irrelevant to our estimate of the critical exponent.

Note that successes and failures make contributions of the same order to the total work. (The reader familiar with ref. 3, Section 3.4, will note that this analysis is virtually identical to the corresponding analysis for the pivot algorithm, with p replaced by $\gamma - 1$.)

Combining (3.2) and (3.11), we find that the mean work for one iteration of the complete practical algorithm—n_{piv} pivot attempts on each walk, followed by one join-and-cut attempt—is

$$\begin{aligned} E(\text{total work}) &= 2n_{\text{piv}} E(\text{pivot work}) + E(\text{join-and-cut work}) \\ &\sim n_{\text{piv}} N_{\text{tot}}^{1-p} + N_{\text{tot}}^{2-\gamma} \end{aligned} \tag{3.12}$$

Now empirically we have $p \leqslant \gamma - 1$, with strict inequality in all dimensions $d < 4$.[3,27] Therefore, *the CPU time of the complete join-and-cut algorithm is dominated by the pivot moves*, even if $n_{\text{piv}} = 1$. That is,

$$E(\text{total work}) \sim n_{\text{piv}} N_{\text{tot}}^{1-p} \tag{3.13}$$

3.2. Improved Method

The process of checking a join-and-cut move for self-avoidance can be speeded up if we happen to have available an up-to-date hash table for the walk that is proposed to grow in length. Then we can simply insert into this.hash table the points of the M-step subwalk, again working outward from the joining point. If no intersection is detected, then the join-and-cut move is successful, and the hash table remains up-to-date. If an intersection is detected, then the join-and-cut move is unsuccessful, and we restore the hash table to its original condition by deleting the temporarily inserted entries (of which we have kept a linear list for this purpose). This refinement of the algorithm does not change the scaling behavior of the computational complexity, but it does improve the overall constant.

The combined pivot and join-and-cut algorithm is therefore implemented as follows:

1. For each walk we maintain *two* hash tables: at any given time one is "active" and the other is "scratch" (a flag keeps track of which is which). With each hash table we also maintain a linear list of its entries, in the order they were inserted; this facilitates clearing the table. A "scratch" hash table is empty except when it is in use. An "active" hash table can be in either of two states, "up-to-date" or "blank," again indicated by a flag. A "blank" hash table is empty.

2. The pivot subroutine uses the "scratch" hash table to carry out the self-avoidance check. If the pivot move is successful, then the active-scratch flag is flipped, and the new "active" hash table is labeled "up-to-date." If

the pivot move is unsuccessful, nothing is done. In either case, the (new) "scratch" hash table is cleared.

3. The join-and-cut subroutine first checks whether the "active" hash table for the walk that is proposed to grow in length is "up-to-date" or not.

3a. If the "active" hash table is "up-to-date," then the points of the M-step subwalk are inserted into this hash table. If an intersection is detected, then the join-and-cut move is unsuccessful, and we restore the hash table to its original "up-to-date" condition by deleting the temporarily inserted entries. If no intersection is detected, then the join-and-cut move is successful, and the hash table remains "up-to-date"; but the "active" hash table for the *other* walk has now become incorrect, so it is cleared and declared "blank."

3b. If the "active" hash table is "blank," then the points of both the M-step subwalk and the N-step walk are inserted into this hash table, as discussed initially (working outward from the joining point). If an intersection is detected, then the join-and-cut move is unsuccessful, and we restore the hash table to its original "blank" condition by clearing it. If no intersection is detected, then the join-and-cut move is successful, and this hash table is now "up-to-date"; but the "active" hash table for the *other* walk is now incorrect, so it is cleared and declared "blank."

Thus, the "active" hash table for a given walk (ω_1 or ω_2) becomes "up-to-date" each time that walk undergoes a successful pivot move; it becomes "blank" each time that walk *decreases* in length in a successful join-and-cut move.

It is interesting to estimate the probability that in step 3 the "active" hash table will be found to be "blank." This happens whenever the most recent successful join-and-cut move in which the walk in question decreased in length has been followed by an uninterrupted succession of pivot failures. Under the approximation that all these events are independent, we get

$$\begin{aligned}\text{Prob(“blank”)} &= \sum_{m=1}^{\infty} \frac{f_{\text{join}}}{2}\left(1-\frac{f_{\text{join}}}{2}\right)^{m-1} (1-f_{\text{pivot}})^{n_{\text{piv}}m} \\ &= \frac{\frac{1}{2}f_{\text{join}}(1-f_{\text{pivot}})^{n_{\text{piv}}}}{1-(1-\frac{1}{2}f_{\text{join}})(1-f_{\text{pivot}})^{n_{\text{piv}}}} \\ &\sim \frac{f_{\text{join}}}{\max[f_{\text{join}},\, 1-(1-f_{\text{pivot}})^{n_{\text{piv}}}]} \end{aligned} \tag{3.14}$$

Since $f_{\text{join}} \sim N_{\text{tot}}^{-(\gamma-1)}$ while $f_{\text{pivot}} \sim N_{\text{tot}}^{-p}$ with (empirically) $p \leqslant \gamma - 1$, we have

$$\text{Prob(“blank”)} \sim N_{\text{tot}}^{-(\gamma-1-p)} \tag{3.15}$$

for any fixed $n_{\text{piv}} \geqslant 1$. In particular, in dimension $d < 4$ we have (empirically) $p < \gamma - 1$, so as $N_{\text{tot}} \to \infty$, the hash table is nearly certain to be "up-to-date."

The algorithm described in this subsection is the one that we have implemented.

4. STATISTICAL METHODS

In this section we show how the critical exponent γ can be estimated from the Monte Carlo data produced by the join-and-cut algorithm. The relevant statistical method is *maximum-likelihood estimation*,[19] which is well known to be an *optimal* statistical method (in the large-sample limit) for parametric-estimation problems. Our use of maximum-likelihood estimation parallels closely that of ref. 9.

4.1. Basic Method

The join-and-cut algorithm produces a (correlated) sequence of pairs $(\omega_1, \omega_2) \in \mathscr{T}_{N_{\text{tot}}}$ coming from the equal-weight probability distribution (1.3). In particular, the random variable $N_1 \equiv |\omega_1|$ has the distribution

$$\bar{\pi}(N_1) = \frac{c_{N_1} c_{N_{\text{tot}} - N_1}}{Z(N_{\text{tot}})} \qquad (0 \leqslant N_1 \leqslant N_{\text{tot}}) \tag{4.1}$$

where $Z(N_{\text{tot}})$ is defined in (1.4). From the observed statistics of N_1 we can make inferences about the unknown constants $\{c_N\}$, and in particular about the critical exponent γ defined in (1.5). Of course, in a strict sense this is impossible, since (1.5) is an asymptotic statement valid as $N \to \infty$, while a Monte Carlo experiment concerns only a finite range of N. Thus, it is necessary to adopt additional *assumptions* about the behavior of the $\{c_N\}$ if we are to proceed further. We begin by making the simplifying assumption that the relation

$$c_N = \mu^N N^{\gamma - 1} a_0 \tag{4.2}$$

is *exact* whenever N is greater than or equal to some cutoff value N_{min} (which we can choose later); here μ, γ, and a_0 are unknown constants. Of course, this assumption is manifestly false: (4.2) is only an *approximation* which gets better and better as N gets large. Thus, our estimate of γ will be afflicted with a *systematic* error arising from corrections to (4.2); it adds to the purely *statistical* error inherent in any Monte Carlo experiment (and which we shall calculate forthwith). We discuss corrections to scaling in Section 4.3.

So let us assume for now that (4.2) is exact for $N \geqslant N_{\min}$. Then the probability distribution of N_1, *conditional on it being in the range from* $N_{\min}$ *to* $N_{\text{tot}} - N_{\min}$, is

$$\operatorname{Prob}(N_1 \mid N_{\min} \leqslant N_1 \leqslant N_{\text{tot}} - N_{\min}) = \frac{[N_1(N_{\text{tot}} - N_1)]^{\gamma-1}}{\mathscr{Z}(N_{\text{tot}}, N_{\min}, \gamma)} \tag{4.3}$$

where

$$\mathscr{Z}(N_{\text{tot}}, N_{\min}, \gamma) = \sum_{N_1 = N_{\min}}^{N_{\text{tot}} - N_{\min}} [N_1(N_{\text{tot}} - N_1)]^{\gamma-1} \tag{4.4}$$

Note that μ and a_0 have dropped out of this formula.

Now suppose that we have a random sample $N_1^{(1)}, \ldots, N_1^{(n)}$, which for the moment we assume consists of *independent* data. (We discuss the treatment of autocorrelated data in Section 4.2.) The likelihood ($\equiv$probability) for such a sequence is simply the product of the individual likelihoods (4.3):

$$\text{likelihood} = \prod_{\substack{1 \leqslant i \leqslant n \\ N_{\min} \leqslant N_1^{(i)} \leqslant N_{\text{tot}} - N_{\min}}} \frac{[N_1^{(i)}(N_{\text{tot}} - N_1^{(i)})]^{\gamma-1}}{\mathscr{Z}(N_{\text{tot}}, N_{\min}, \gamma)} \tag{4.5}$$

(Here the product is taken only over those i for which $N_{\min} \leqslant N_1^{(i)} \leqslant N_{\text{tot}} - N_{\min}$; the other data points are simply discarded.) The maximum-likelihood estimate $\hat{\gamma}$ is, by definition, that value of γ which, for the given data $N_1^{(1)}, \ldots, N_1^{(n)}$, maximizes the likelihood (4.5). Simple calculus shows that $\hat{\gamma}$ is determined by the condition

$$\langle X \rangle_{\hat{\gamma}} = \langle X \rangle_{\text{obs}} \tag{4.6}$$

where we have defined the random variable

$$X \equiv \log[N_1(N_{\text{tot}} - N_1)] \tag{4.7}$$

and the theoretical and observed mean values

$$\langle f(N_1) \rangle_\gamma \equiv \frac{\{\sum_{N_1 = N_{\min}}^{N_{\text{tot}} - N_{\min}} f(N_1)[N_1(N_{\text{tot}} - N_1)]^{\gamma-1}\}}{\sum_{N_1 = N_{\min}}^{N_{\text{tot}} - N_{\min}} [N_1(N_{\text{tot}} - N_1)]^{\gamma-1}} \tag{4.8}$$

$$\langle f(N_1) \rangle_{\text{obs}} \equiv \frac{\{\sum_{N_{\min} \leqslant N_1^{(i)} \leqslant N_{\text{tot}} - N_{\min}} f(N_1^{(i)})\}}{\sum_{N_{\min} \leqslant N_1^{(i)} \leqslant N_{\text{tot}} - N_{\min}} 1} \tag{4.9}$$

Equation (4.6) is easily solved numerically for $\hat{\gamma}$. By the general theory of maximum-likelihood estimation,(19,29,30) the probability distribution of $\hat{\gamma}$ is asymptotically Gaussian as the sample size n tends to infinity, with mean

$$\langle \hat{\gamma} \rangle_\gamma = \gamma + O\left(\frac{1}{n'}\right) \tag{4.10}$$

and variance

$$\operatorname{var}(\hat{\gamma}) \equiv \langle \hat{\gamma}; \hat{\gamma} \rangle_\gamma = \frac{1}{n' \langle X; X \rangle_\gamma} + O\left(\frac{1}{n'^2}\right) \tag{4.11}$$

where

$$n' \equiv \sum_{N_{\min} \leqslant N_1^{(i)} \leqslant N_{\text{tot}} - N_{\min}} 1 \tag{4.12}$$

is the censored sample size. We have here used the notation

$$\langle A; B \rangle \equiv \langle AB \rangle - \langle A \rangle \langle B \rangle \tag{4.13}$$

Note that $\operatorname{var}(\hat{\gamma})$ depends in principle on the unknown "true" value of γ; but since this dependence is rather weak, and since $\hat{\gamma}$ will be a fairly close estimate of γ (if $n' \gg 1$), it suffices for our purposes to replace γ by the estimated value $\hat{\gamma}$ when attempting to compute error bars.[13]

A very important feature of this method is that *the statistical error bars can be computed prior to performing the Monte Carlo experiment.* Let us do so. We have

$$\begin{aligned} \langle X; X \rangle_\gamma = {} & \sum_{N_1 = N_{\min}}^{N_{\text{tot}} - N_{\min}} X^2 [N_1 (N_{\text{tot}} - N_1)]^{\gamma - 1} \Bigg/ \sum_{N_1 = N_{\min}}^{N_{\text{tot}} - N_{\min}} [N_1 (N_{\text{tot}} - N_1)]^{\gamma - 1} \\ & - \left(\sum_{N_1 = N_{\min}}^{N_{\text{tot}} - N_{\min}} X [N_1 (N_{\text{tot}} - N_1)]^{\gamma - 1} \Bigg/ \sum_{N_1 = N_{\min}}^{N_{\text{tot}} - N_{\min}} [N_1 (N_{\text{tot}} - N_1)]^{\gamma - 1} \right)^2 \end{aligned} \tag{4.14}$$

where X has been defined earlier. These sums can easily be evaluated numerically. Alternatively, for large N_{tot} (which is the regime of interest),

[13] More rigorously, one would limit oneself to some interval $[\gamma_1, \gamma_2]$ in which the true γ value is *assumed* to lie, and compute the *worst possible* error bars subject to that assumption; this defines a rigorous confidence interval for γ subject to the assumption that $\gamma \in [\gamma_1, \gamma_2]$.

the sums can be approximated by integrals: letting $y = N_1/N_{\text{tot}}$ and $y_{\min} = N_{\min}/N_{\text{tot}}$, we have

$$\begin{aligned}\langle X; X\rangle_\gamma &\approx \frac{\int_{y_{\min}}^{1-y_{\min}} dy \log^2[y(1-y)][y(1-y)]^{\gamma-1}}{\int_{y_{\min}}^{1-y_{\min}} dy\, [y(1-y)]^{\gamma-1}} \\ &\quad - \left(\frac{\int_{y_{\min}}^{1-y_{\min}} dy \log[y(1-y)][y(1-y)]^{\gamma-1}}{\int_{y_{\min}}^{1-y_{\min}} dy\, [y(1-y)]^{\gamma-1}}\right)^2 \qquad (4.15\text{a}) \\ &= \frac{d^2}{d\gamma^2} \log \int_{y_{\min}}^{1-y_{\min}} dy\, [y(1-y)]^{\gamma-1} \qquad (4.15\text{b})\end{aligned}$$

A numerical evaluation of (4.15a) yields the results shown in Table II. For example, at $\gamma = 1.3$ and $y_{\min} = 0.10$, we have $\langle X; X\rangle \approx 0.07$; this means that $\operatorname{var}(\hat{\gamma}) \approx 1/0.07n' \approx 14/n'$, where n' is the number of "effectively independent" censored samples. From Table II one can thus estimate the sample size needed to achieve a specified accuracy for γ.

Remark. The integral (4.15b) is an incomplete beta function. When $y_{\min} \to 0$ it becomes an ordinary beta function, and

$$\langle X; X\rangle_\gamma = \frac{d^2}{d\gamma^2}[2 \log \Gamma(\gamma) - \log \Gamma(2\gamma)] \qquad (4.16)$$

Table II. $\langle X; X\rangle$ As a Function of γ (from 1.0 to 1.5) and $y_{\min} \equiv N_{\min}/N_{\text{tot}}$ (from 0.0 to 0.15)

$y_{\min}$	$\gamma = 1.0$	$\gamma = 1.1$	$\gamma = 1.2$	$\gamma = 1.3$	$\gamma = 1.4$	$\gamma = 1.5$
0.0	0.710132	0.574867	0.474144	0.397279	0.337385	0.289868
0.01	0.432901	0.386997	0.345751	0.308956	0.276310	0.247457
0.02	0.327874	0.300873	0.275664	0.252301	0.230781	0.211059
0.03	0.260792	0.243105	0.226233	0.210248	0.195192	0.181084
0.04	0.212792	0.200552	0.188710	0.177323	0.166433	0.156069
0.05	0.176410	0.167646	0.159082	0.150758	0.142708	0.134958
0.06	0.147836	0.141414	0.135092	0.128897	0.122855	0.116985
0.07	0.124847	0.120062	0.115324	0.110652	0.106065	0.101578
0.08	0.106025	0.102414	0.098824	0.095266	0.091755	0.088300
0.09	0.090410	0.087660	0.084916	0.082187	0.079481	0.076807
0.10	0.077323	0.075214	0.073103	0.070998	0.068903	0.066825
0.11	0.066268	0.064642	0.063010	0.061379	0.059751	0.058132
0.12	0.056870	0.055610	0.054345	0.053077	0.051810	0.050545
0.13	0.048839	0.047861	0.046878	0.045890	0.044902	0.043913
0.14	0.041951	0.041190	0.040424	0.039654	0.038882	0.038109
0.15	0.036022	0.035430	0.034833	0.034233	0.033630	0.033026

Moreover, for $\gamma=1$ and $\gamma=3/2$, the integrals (4.15a) with $y_{\min}=0$ are elementary, and we find

$$\langle X; X\rangle_{\gamma=1}=4-\frac{\pi^2}{3}\approx 0.710132 \tag{4.17a}$$

$$\langle X; X\rangle_{\gamma=3/2}=\frac{\pi^2}{3}-3\approx 0.289868 \tag{4.17b}$$

Unfortunately, these values are good approximations only if $y_{\min}\equiv N_{\min}/N_{\text{tot}}$ is *very* small ($\lesssim 0.01$), as can be seen from Table II.

4.2. Autocorrelated Data

In Section 4.1 we have developed the maximum-likelihood theory under the simplifying assumption that the sample $N_1^{(1)},..., N_1^{(n)}$ is a sequence of *independent* random variables. But we know that in fact the join-and-cut algorithm generates a *correlated* sequence. How does this autocorrelation affect the maximum-likelihood theory?

It turns out[31-33] that the maximum-likelihood method is still valid, but the variance of $\hat{\gamma}$ is now

$$\operatorname{var}(\hat{\gamma})=\frac{\operatorname{var}(\bar{X})}{\langle X; X\rangle_\gamma^2}+O\left(\frac{1}{n'^2}\right) \tag{4.18a}$$

$$=\frac{2\tau_{\text{int},X}}{n'\langle X; X\rangle_\gamma}+O\left(\frac{1}{n'^2}\right) \tag{4.18b}$$

where $\langle X; X\rangle_\gamma$ is as before, and the variance of the sample mean $\bar{X}$ must be determined through the appropriate time-series analysis.[3,20] [If the samples were independent, one would have $2\tau_{\text{int},X}=1$, and (4.18) would reduce to (4.11).]

Let us discuss in a bit more detail how we handle the "censoring" $N_{\min}\leqslant N_1\leqslant N_{\text{tot}}-N_{\min}$ in the context of autocorrelated data. One approach is to realize that the censored sample mean $\bar{X}^{\text{cens}}$ is simply a ratio of two uncensored means:

$$\bar{X}^{\text{cens}}=\frac{(1/n)\sum_{i=1}^n \theta^{(i)}X^{(i)}}{(1/n)\sum_{i=1}^n \theta^{(i)}}\equiv\frac{\overline{\theta X}^{\text{uncens}}}{\bar{\theta}^{\text{uncens}}} \tag{4.19}$$

where θ denotes the indicator function of the event $N_{\min}\leqslant N_1\leqslant N_{\text{tot}}-N_{\min}$, i.e.,

$$\theta^{(i)}=\begin{cases}1 & \text{if } N_{\min}\leqslant N_1^{(i)}\leqslant N_{\text{tot}}-N_{\min}\\ 0 & \text{otherwise}\end{cases} \tag{4.20}$$

We can then use the standard formula for the variance of a ratio (which is valid in the large-sample limit),

$$\operatorname{var}\left(\frac{A}{B}\right)=\frac{\langle A\rangle^2}{\langle B\rangle^2}\operatorname{var}\left(\frac{A}{\langle A\rangle}-\frac{B}{\langle B\rangle}\right) \tag{4.21}$$

with $A=\overline{\theta X}^{\text{uncens}}$ and $B=\bar{\theta}^{\text{uncens}}$. The latter variance can be estimated by the usual time-series analysis[3,20] applied to the uncensored time series for the observable $C\equiv\theta X/\langle\theta X\rangle-\theta/\langle\theta\rangle$, or more precisely

$$C^{(i)}=\theta^{(i)}X^{(i)}/\overline{\theta X}^{\text{uncens}}-\theta^{(i)}/\bar{\theta}^{\text{uncens}} \tag{4.22}$$

This variance is determined from the autocorrelation time $\tau_{\text{int},C}$ in the uncensored time series.

An alternate approach is to work directly with the "censored time series" $\tilde{X}^{(1)},\dots,\tilde{X}^{(n')}$ defined by

$$\tilde{X}^{(j)}=X^{(T_j)} \tag{4.23}$$

where $T_1<T_2<\cdots<T_{n'}$ are the times i for which $N_{\min}\leqslant N_1^{(i)}\leqslant N_{\text{tot}}-N_{\min}$. We emphasize that the $\{T_j\}$ are *random variables*, as is the censored sample size n'. The censored time series is a stationary stochastic process (and indeed, the underlying censored process on the state space $\mathscr{T}_{N_{\text{tot}}}$ is a Markov chain), though of course "time" has a different meaning here than it had in the uncensored process. It is natural, therefore, to analyze this time series by the usual techniques[3,20]; these techniques yield an error bar on $\bar{\tilde{X}}=\bar{X}^{\text{cens}}$, based on the autocorrelation time $\tau_{\text{int},\tilde{X}}$ in the censored time series.

We thus have two alternate methods for determining the error bar on $\bar{X}^{\text{cens}}$, so we have to ask: which one is correct? The first method is obviously unobjectionable. The second method, however, is based on a slight cheat, because the sample size n' is a random variable and we are treating it as if it were a deterministic quantity. Nevertheless, it turns out that the central limit theorem for Markov chains can be generalized to the case in which the upper time limit is a *random variable* that tends to infinity in a suitable sense, and the limiting variance for the sample mean is the *same* (up to the obvious time-rescaling factor $\langle\theta\rangle$) whether the upper limit is deterministic or random.[14] We believe that these facts can be used to

[14] See, for example, ref. 34, Theorem 20.3; ref. 35, Corollary 4.2, Theorem 4.2, and Section 5; and ref. 36, Theorem 1 and its proof. For related material, see also ref. 37 and ref. 38, Section 7.4, and the references cited there. The important point is that the sequence $(T_j/j)_{j\geqslant1}$ is a *subsequence* of the sequence $(n/\sum_{i=1}^n\theta_i)_{n\geqslant1}$; and since the strong law of large numbers (=ergodic theorem) for Markov chains guarantees that $\lim_{n\to\infty}\sum_{i=1}^n\theta_i/n=\langle\theta\rangle$ almost surely, it follows that $\lim_{j\to\infty}T_j/j=1/\langle\theta\rangle$ almost surely.

prove the asymptotic (large-sample) equivalence of the two methods. In any case, we have tried both methods on our data, and we obtain estimates of $\bar{X}$ that are *equal* within error bars.

Let us also mention how to combine data from runs at different values of N_{tot}. The simplest approach is to analyze the data separately for each N_{tot}, and then to construct a weighted average of the resulting estimates $\hat{\gamma}$, with weights inversely proportional to the estimated variances $\text{var}(\hat{\gamma})$ taken from (4.18):

$$\hat{\gamma}^{\text{overall}} = \frac{\sum_j w_j \hat{\gamma}_j}{\sum_j w_j} \tag{4.24}$$

with $w_j = 1/\text{var}(\hat{\gamma}_j)$. An equivalent approach (when the individual-run sample sizes are all large) is to solve the weighted likelihood equations

$$\sum_j w'_j \langle X \rangle_{N_{\text{tot},j}, N_{\text{min},j}, \hat{\gamma}^{\text{overall}}} = \sum_j w'_j \bar{X}_j^{\text{cens}} \tag{4.25}$$

where the weights w'_j are taken proportional to the number of "effectively independent" censored samples, i.e.,

$$w'_j = \frac{n'_j}{\tau_{\text{int},\bar{X},j}} \tag{4.26}$$

The proof of equivalence is simple: write $\hat{\gamma}^{\text{overall}} = \hat{\gamma}_j + \varepsilon_j$ and expand each term on the left-hand side of (4.25) using

$$\langle X \rangle_{N_{\text{tot},j}, N_{\text{min},j}, \hat{\gamma}^{\text{overall}}} = \langle X \rangle_{N_{\text{tot},j}, N_{\text{min},j}, \hat{\gamma}_j} + \varepsilon_j \langle X; X \rangle_{N_{\text{tot},j}, N_{\text{min},j}, \hat{\gamma}_j} + O(\varepsilon_j^2) \tag{4.27}$$

Now, by definition of $\hat{\gamma}_j$, we have

$$\langle X \rangle_{N_{\text{tot},j}, N_{\text{min},j}, \hat{\gamma}_j} = \bar{X}_j^{\text{cens}} \tag{4.28}$$

for each j. It follows that $\sum_j w''_j \varepsilon_j = 0$, where

$$w''_j \equiv w'_j \langle X; X \rangle_{N_{\text{tot},j}, N_{\text{min},j}, \hat{\gamma}_j}$$

But $w''_j = \text{const} \times w_j$, so this is precisely (4.24).

We can now pose and solve an optimization problem: For fixed N_{min}, what is the optimal choice of N_{tot} if the CPU time per "independent" sample scales as $\sim N_{\text{tot}}^k$? [Combining (2.46), (3.13), and (4.18b), we expect $k = 1 - p + (r_X + a_X s_X) = \gamma - p + a_X s_X$ for the algorithm with $n_{\text{piv}} = 1$.] The goal is to minimize the variance of $\hat{\gamma}$ per unit CPU time, hence to maximize $\langle X; X \rangle_{\gamma, y_{\text{min}}} \times y_{\text{min}}^k$. The optimal value $y_{\text{min,opt}}$ as a function of γ and k is shown in Table III. For $d = 2$ we have $\gamma = 1.34375$[39,40] and $k \approx 1.5$ (see

Table III. Value of $y_{\min}$ Maximizing $\langle X; X\rangle_{\gamma, y_{\min}} \times y_{\min}^k$, for Various γ and k

k	$\gamma = 1.0$	$\gamma = 1.1$	$\gamma = 1.2$	$\gamma = 1.3$	$\gamma = 1.4$	$\gamma = 1.5$
1.0	0.057268	0.060042	0.062836	0.065635	0.068428	0.071208
1.1	0.065154	0.067848	0.070561	0.073279	0.075994	0.078696
1.2	0.073026	0.075623	0.078239	0.080863	0.083484	0.086094
1.3	0.080836	0.083325	0.085834	0.088353	0.090872	0.093382
1.4	0.088545	0.090921	0.093319	0.095728	0.098139	0.100545
1.5	0.096125	0.098387	0.100672	0.102969	0.105271	0.107570
1.6	0.103556	0.105705	0.107876	0.110063	0.112255	0.114448
1.7	0.110823	0.112860	0.114922	0.116999	0.119084	0.121171
1.8	0.117915	0.119845	0.121800	0.123771	0.125751	0.127735
1.9	0.124826	0.126654	0.128505	0.130374	0.132254	0.134140
2.0	0.131552	0.133282	0.135036	0.136808	0.138591	0.140381

Section 5.1), so $y_{\min,\mathrm{opt}} \approx 0.1$. For $d = 3$ we have $\gamma \approx 1.16$,[41] and we conjecture $k \approx 1.2$–1.3 (see Section 6), so $y_{\min,\mathrm{opt}} \approx 0.08$. Finally, for $d \geqslant 4$ we have $\gamma = 1$ and we conjecture $k = 1$ (see Section 6), so $y_{\min,\mathrm{opt}} \approx 0.06$.

4.3. Corrections to Scaling

We now return to the problem of corrections to scaling. Clearly, (1.5) is only the leading term in the asymptotic expansion of c_N for large N; the renormalization group predicts[42] that the actual behavior is

$$c_N \sim \mu^N N^{\gamma-1} \left(a_0 + \frac{a_1}{N} + \frac{a_2}{N^2} + \cdots + \frac{b_0}{N^{\Delta_1}} \right.$$
$$\left. + \frac{b_1}{N^{\Delta_1+1}} + \cdots + \frac{c_0}{N^{\Delta_2}} + \frac{c_1}{N^{\Delta_2+1}} + \cdots \right) \tag{4.29}$$

where $\Delta_1 < \Delta_2 < \cdots$ are correction-to-scaling exponents. Here, in addition to analytic corrections of the form a_k/N^k, there are nonanalytic corrections of the general form $\mathrm{const}/N^{k_1\Delta_1 + k_2\Delta_2 + \cdots + l}$, where $k_1, k_2,\ldots$, and l are nonnegative integers. The exponents $\Delta_1, \Delta_2,\ldots$ are believed to be universal among lattices of a given dimension d, while the amplitudes $a_0, a_1,\ldots$, $b_0, b_1,\ldots$, $c_0, c_1,\ldots$ are nonuniversal.

The maximum-likelihood analysis described above is based on the assumption that (4.2) is exact for $N \geqslant N_{\min}$; if (4.29) is correct, then this assumption is in error by an amount of order $1/N_{\min}^{\Delta}$, where $\Delta \equiv \min(\Delta_1, 1)$. Thus we expect that the estimates of γ derived using (4.2) have likewise a systematic error of this order (as well as higher-order corrections). One should therefore perform this analysis for a variety of values of

$N_{\min}$ and attempt an extrapolation to $N_{\min} = \infty$. Of course, such an extrapolation is difficult, because the statistical error bars on $\hat{\gamma}$ grow rapidly with $N_{\min}$. Thus, some uncertainty about the "correct" $N_{\min} = \infty$ limit of the central value $\hat{\gamma}$ will inevitably remain. A subjective estimate of this residual uncertainty should therefore be reported as a possible *systematic error* induced by unaccounted-for corrections to scaling; it adds to the purely *statistical error* computed from (4.11) and embodied in the statistical error bars at each fixed $N_{\min}$.

As a further consistency check, one can perform the foregoing analysis with (4.2) replaced by

$$c_N = \mu^N N^{\gamma - 1} \left(1 + \frac{\tilde{a}_1}{N^{\Delta}}\right) a_0 \tag{4.30}$$

Here Δ and $\tilde{a}_1$ are fixed constants, which are supposed to be guesses for the leading correction-to-scaling exponent and amplitude. We expect[9,43] that the estimates $\hat{\gamma}$ will be "flattest" as a function of $N_{\min}$ when Δ and $\tilde{a}_1$ are given their correct values: in this way, the leading correction-to-scaling term is canceled, and the systematic error is dominated by the next-to-leading correction term. This suggests that one should try a variety of values of Δ and $\tilde{a}_1$ and select the ones that exhibit the weakest dependence on $N_{\min}$. Unfortunately, the validity of the "flatness criterion" is guaranteed only for "sufficiently large" $N_{\min}$, and in practice it may not be possible to go to such a large $N_{\min}$ without incurring a *statistical* error so large as to mask completely the corrections to scaling. In any case, we do our best to apply the "flatness criterion," and we report the residual uncertainty as a systematic error.

For further discussion of corrections to scaling, see ref. 9, pp. 503–505, and ref. 43.

5. NUMERICAL RESULTS

We performed runs of the join-and-cut algorithm for SAWs on the square lattice ($d = 2$) at a variety of values of N_{tot} and n_{piv}: see Table IV. (We actually simulated the variant algorithm with $N^*_{\min} = 1$: see Remark 2 in Section 2.1.) We took data once every two iterations[15]; all autocorrelation times reported here are measured in units of two iterations. We analyzed the data using standard procedures of statistical time-series analysis; more details can be found in ref. 3, Appendix C. We used in all cases a self-consistent truncation window of width $5\tau_{\text{int},\mathcal{A}}$.

[15] Slightly better statistics would have been obtained by measuring every iteration, but lack of disk space made this inconvenient.

Table IV. Number of Iterations Performed (in 10^5) As a Function of N_{tot} and n_{piv}

N_{tot}	$n_{\text{piv}}=1$	$=2$	$=5$	$=10$	$=20$	$=40$	$=80$	$=160$
500	46	4	2	2	2	2	4	4
1000	46	4	2	2	4	4	4	
2000	105	4	2	2	2	4		
4000	63	4	2	2	2	3		
8000	25	4	2	2	2	4		

All error bars reported in this section are, except where otherwise specified, *one* standard deviation (i.e., 68% confidence limits) and include *statistical error only*. It should be borne in mind that the systematic error arising from corrections to scaling may be much larger than the statistical errors.

5.1. Acceptance Fraction and Dynamic Critical Behavior

In Table V we report the acceptance fractions $f \equiv f(\text{join-and-cut})$ and $f(\text{pivot})$ as a function of N_{tot}. From (2.30b) and (2.32), we expect that $f \sim N_{\text{tot}}^{-(\gamma-1)}$, where $\gamma - 1 = 11/32 = 0.34375$ in $d=2$,[39,40] and $f(\text{pivot}) \sim N_{\text{tot}}^{-p}$, where $p \approx 0.19$ in $d=2$.[3] A least-squares fit to Table V yields the exponents

$$\text{join-and-cut:} \quad \begin{cases} 0.333 \pm 0.002 \ (\chi^2 = 3.49, \text{level} = 32\,\%) \\ \quad \text{using all data points} \\ 0.338 \pm 0.003 \ (\chi^2 = 0.64, \text{level} = 73\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 1000 \\ 0.338 \pm 0.006 \ (\chi^2 = 0.63, \text{level} = 43\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 2000 \end{cases} \tag{5.1}$$

$$\text{pivot:} \quad \begin{cases} 0.1967 \pm 0.0004 \ (\chi^2 = 11.41, \text{level} = 1\,\%) \\ \quad \text{using all data points} \\ 0.1963 \pm 0.0006 \ (\chi^2 = 10.82, \text{level} = 0.5\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 1000 \\ 0.1935 \pm 0.0011 \ (\chi^2 = 0.63, \text{level} = 43\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 2000 \end{cases} \tag{5.2}$$

Here "level" is the probability that χ^2 would exceed the observed value, assuming the correctness of the power-law model and of the raw-data error

Table V. Acceptance Fractions for the Two Types of Moves[a]

N_{tot}	Join-and-cut	Pivot
500	0.2323 ± 0.0005	0.3531 ± 0.0002
1000	0.1853 ± 0.0005	0.3082 ± 0.0002
2000	0.1468 ± 0.0005	0.2682 ± 0.0002
4000	0.1157 ± 0.0006	0.2347 ± 0.0002
8000	0.0922 ± 0.0008	0.2050 ± 0.0003

[a] Error bars are one standard deviation.

bars; it can serve as a test of goodness of fit. An abnormally large value of χ^2 (say, a level less than 5%) may indicate *either* that the pure power-law Ansatz is incorrect (e.g., due to corrections to scaling) or else that the claimed error bars on the raw data are too small; further investigation would be necessary to determine which of these is the true cause. An abnormally small value of χ^2 (say, a level greater than 95%) probably indicates that the claimed error bars on the raw data are too large. The fits (5.1)–(5.2) indicate good agreement with the predicted exponents, up to possible systematic errors of magnitude ≈ 0.005–0.01 due to corrections to scaling.

In Table VI we report the autocorrelation time $\tau_{\text{int},X}$ for the observable $X \equiv \log[N_1(N_{\text{tot}} - N_1)]$, as a function of N_{tot} and n_{piv}. Plots of $\tau_{\text{int},X}$ versus $1/n_{\text{piv}}$ for each fixed N_{tot} are remarkably linear (except that for $N_{\text{tot}} = 500$ we must discard $n_{\text{piv}} = 1, 2$), in agreement with the hand-waving prediction (2.39). The linear least-squares extrapolation to $1/n_{\text{piv}} = 0$ is reported in the final column of Table VI (the error bar denotes statistical error only: possible systematic errors in the extrapolation are ignored). A least-squares fit of the extrapolated $\tau_{\text{int},X}(n_{\text{piv}} = \infty)$ values versus N_{tot} yields the exponent estimates

$$r_X = \begin{cases} 0.400 \pm 0.007 \ (\chi^2 = 4.30, \text{ level} = 23\,\%) \\ \quad \text{using all data points} \\ 0.396 \pm 0.010 \ (\chi^2 = 3.87, \text{ level} = 14\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 1000 \\ 0.373 \pm 0.020 \ (\chi^2 = 2.02, \text{ level} = 16\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 2000 \end{cases} \tag{5.3}$$

The systematic downward trend in the exponent appears to indicate the presence of a strong correction-to-scaling term. (The lack of improvement

Table VI. Autocorrelation Time $\tau_{\text{int},\mathcal{X}}$ As a Function of N_{tot} and n_{piv}, Measured in Units of Two Iterations[a]

N_{tot}	$n_{\text{piv}} = 1$	$= 2$	$= 5$	$= 10$	$= 20$	$= 40$	$= 80$	$= 160$	$= \infty$ (ext.)
500	2.704(0.044)	2.188(0.032)	1.930(0.038)	1.725(0.032)	1.653(0.030)	1.562(0.028)	1.582(0.028)	1.520(0.026)	1.529(0.016)
1000	4.135(0.084)	3.116(0.055)	2.441(0.054)	2.254(0.048)	2.182(0.032)	2.010(0.040)	2.034(0.041)		2.025(0.020)
2000	6.348(0.160)	4.735(0.103)	3.345(0.087)	3.280(0.084)	2.870(0.069)	2.833(0.067)			2.743(0.043)
4000	10.296(0.330)	7.085(0.189)	4.595(0.139)	4.285(0.125)	3.737(0.102)	3.672(0.100)			3.450(0.066)
8000	16.747(0.685)	9.971(0.315)	7.686(0.301)	5.790(0.197)	5.097(0.163)	4.948(0.156)			4.640(0.107)

[a] Standard deviation is shown in parentheses. Last column is a linear least-squares extrapolation to $1/n_{\text{piv}} = 0$; error bar is statistical error only.

in the goodness of fit as points at low N_{tot} are discarded suggests that the corrections are still strong at $N_{\mathrm{tot}}=2000$.) Indeed, from the bound (2.28) it follows *rigorously* that $r_X \leqslant \gamma - 1$ $(=0.34375)$ if one assumes that an Ansatz of the form (2.35) holds with $F_X(\infty)>0$. Therefore, the estimates (5.3) *cannot* have yet reached the asymptotic regime. In any case, the final estimate 0.373 ± 0.020 is already in fair agreement with the prediction $r_X = \gamma - 1 = 0.34375$ [see (2.36)].

Next we fix $r_X = \gamma - 1 = 11/32$ and attempt to fit $\tau_{\mathrm{int},X}$ at *finite* n_{piv} to the scaling Ansatz (2.35). To do this, we plot $\tau_{\mathrm{int},X}/N_{\mathrm{tot}}^{\gamma-1}$ versus $n_{\mathrm{piv}}/N_{\mathrm{tot}}^{s_X}$ and adjust s_X until the points fall roughly on a single curve $F_X(z)$. Reasonable fits are obtained for s_X anywhere between ≈ 0.50 and ≈ 0.75; we therefore estimate

$$s_X = 0.63 \pm 0.13 \tag{5.4}$$

(subjective 68% confidence limits). This is compatible with the prediction $s_X = p + \gamma - 1 \approx 0.54$ [see (2.44)]. In Fig. 2 we show the scaling plots for $s_X = 0.54$ and $s_X = 0.63$.

Next we analyze the scaling function $F_X(z)$ in the two limiting regimes $z \to 0$ and $z \to \infty$ [see (2.45) and (2.39)]. For $z \to 0$, we fit the data to $F_X(z) \sim z^{-a_X}$ by drawing the estimated asymptote as $z \to 0$ on a log-log plot (see Fig. 2); this yields the crude estimates

$$a_X \approx \begin{cases} 0.61 \pm 0.08 & \text{if} \quad s_X = 0.54 \\ 0.56 \pm 0.06 & \text{if} \quad s_X = 0.63 \end{cases} \tag{5.5}$$

However, the curvature seen in Fig. 2 may well continue to yet smaller values of z, in which case the true a_X would be *higher* than these estimates. To extract the behavior as $z \to \infty$ and test the prediction (2.39), we plot $F_X(z)$ versus $1/z$ in Fig. 3. The asymptotic behavior as $1/z \to 0$ is amazingly linear all the way up to $1/z \lesssim 100$–300. Only the six lowest points (correspondingly to $1/z \lesssim 2$–3) seem to deviate systematically from this linear behavior; and since these points correspond to the lowest values of N_{tot} (namely $N_{\mathrm{tot}} = 500, 1000$), this deviation may well be due to corrections to scaling (see Appendix). It is thus conceivable that the conjectured behavior $F_X(z) = F_X(\infty) + \mathrm{const}/z^{\approx 1}$ may hold not only for large z, but in fact for *all* z (at least in some approximate sense). In this case we would have $a_X \approx 1$; and the lower value of a_X found in (5.5) would be attributable to the large additive constant (about 1/4 of the total value) present even at $z \sim 0.01$. In order to test this conjecture, we will need better data, over a larger range of values of z (in both directions), and possibly at larger values of N_{tot}.

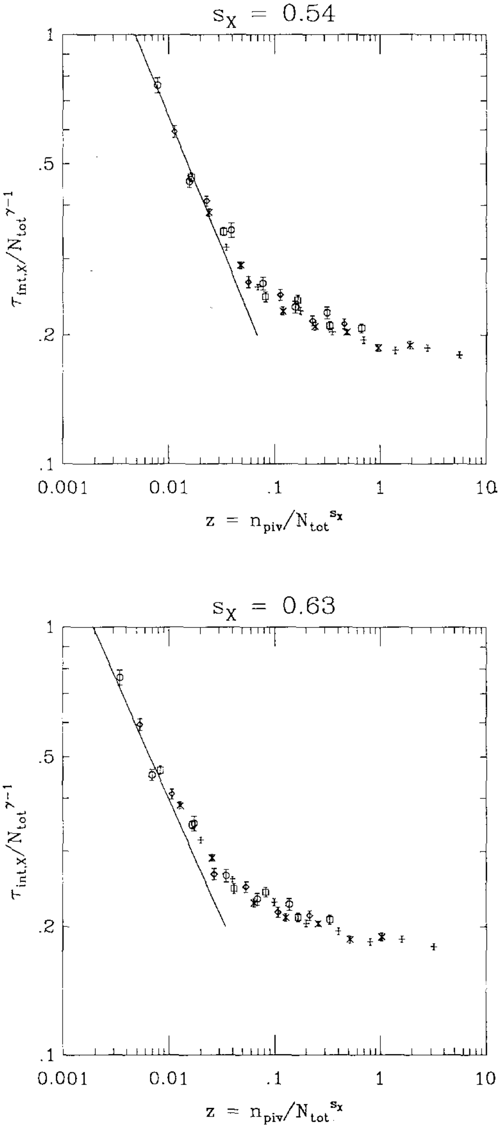

Fig. 2. Log-log plot of $\tau_{\mathrm{int},X}/N_{\mathrm{tot}}^{\gamma-1}$ versus $z \equiv n_{\mathrm{piv}}/N_{\mathrm{tot}}^{s_X}$ for (a) $s_X = 0.54$, (b) $s_X = 0.63$. Points are $N_{\mathrm{tot}} = 500$ (+), 1000 (×), 2000 (□), 4000 (◇), and 8000 (○); note that they fall roughly on a single curve $F_X(z)$. Straight line indicates the estimated asymptote as $z \to 0$.

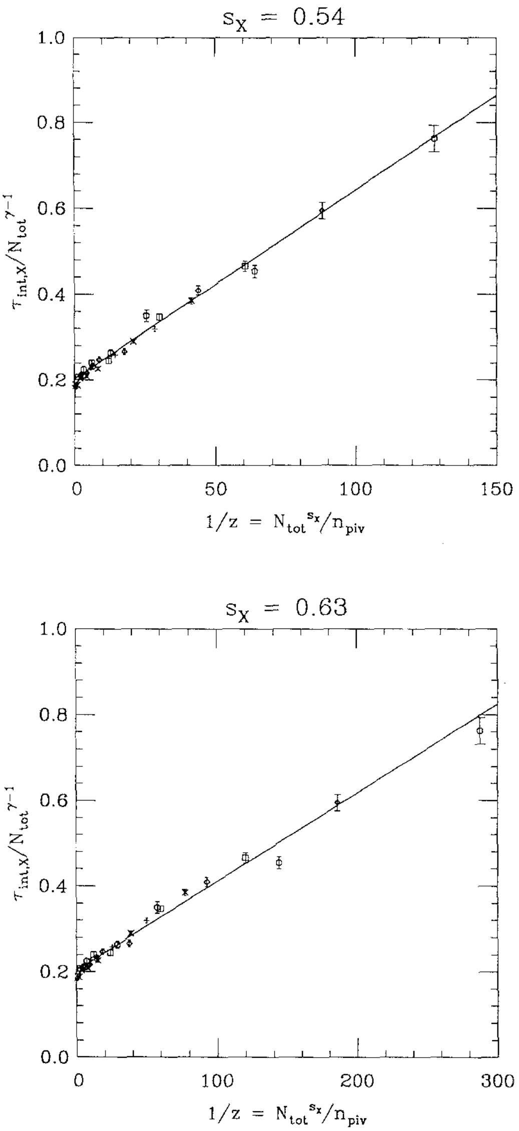

Fig. 3. Plot of $F_X(z) \equiv \tau_{\mathrm{int},X}/N_{\mathrm{tot}}^{\gamma-1}$ versus $1/z \equiv N_{\mathrm{tot}}^{s_X}/n_{\mathrm{piv}}$ for (a) $s_X = 0.54$, (b) $s_X = 0.63$. Points are $N_{\mathrm{tot}} = 500$ (+), 1000 (×), 2000 (□), 4000 (◇), and 8000 (○). Straight line indicates the best linear fit to the data with $N_{\mathrm{tot}} \geqslant 2000$.

Finally, we fit $\tau_{\text{int},X}$ at $n_{\text{piv}}=1$ versus N_{tot} [see (2.46)]: we find

$$q_X = \begin{cases} 0.646 \pm 0.012 \ (\chi^2 = 3.48, \text{level} = 32\,\%) \\ \quad \text{using all data points} \\ 0.667 \pm 0.019 \ (\chi^2 = 1.41, \text{level} = 49\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 1000 \\ 0.699 \pm 0.033 \ (\chi^2 = 0.001, \text{level} = 97\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 2000 \end{cases} \tag{5.6}$$

The estimated exponent shows a clear rise as a function of N_{tot}, indicating a crossover between the exponent $r_X = \gamma - 1 = 0.34375$ valid for $N_{\text{tot}} \lesssim n_{\text{piv}}^{1/s_X} \sim n_{\text{piv}}^{\approx 1.8}$ and the exponent q_X valid for $N_{\text{tot}} \to \infty$ at fixed n_{piv}. This suggests that q_X is *at least* 0.699 ± 0.033 and quite possibly larger. In any case, this estimate is in good agreement with the prediction

$$q_X = \gamma - 1 + a_X s_X = \begin{cases} 0.68 \pm 0.04 & \text{if} \quad s_X = 0.54 \\ 0.70 \pm 0.03 & \text{if} \quad s_X = 0.63 \end{cases} \tag{5.7}$$

5.2. Estimates of γ

We computed maximum-likelihood estimates of the critical exponent γ defined by

$$c_N \sim \mu^N N^{\gamma-1} \left(1 + \frac{\tilde{a}_1}{N^\Delta}\right) \qquad \text{for} \quad N \geqslant N_{\text{min}} \tag{5.8}$$

where Δ is a guess for the leading correction-to-scaling exponent $\Delta \equiv \min(\Delta_1, 1)$, and $\tilde{a}_1$ is a guess for the corresponding amplitude (see Section 4.3). We used $\Delta = 11/16$ and $\Delta = 1$, together with a variety of values of N_{min} and $\tilde{a}_1$. Here $\Delta_1 = 11/16$ is the most likely conformal-invariance prediction for the correction-to-scaling exponent in the two-dimensional SAW,(44,45) while $\Delta = 1$ is the most plausible behavior in case the non-analytic correction-to-scaling is weak or absent. We performed this analysis separately for each run [i.e., for each pair $(N_{\text{tot}}, n_{\text{piv}})$], and then combined the estimates according to (4.24). The final results are shown in Table VII. It is somewhat difficult to apply the "flatness criterion"(9,43): for any $\tilde{a}_1$ in the range $0.5 \lesssim \tilde{a}_1 \lesssim 2.3$ $(\Delta = 1)$ or $0.3 \lesssim \tilde{a}_1 \lesssim 0.9$ $(\Delta = 11/16)$, we obtain flatness within error bars for $N_{\text{min}} \gtrsim 80$. (We define "flatness" as minimizing the spread among the various estimates for $N_{\text{min}} \geqslant 80$.) Our final estimates for γ are

$$\gamma = \begin{cases} 1.3523 \pm 0.0074 \pm 0.0060 & \text{assuming} \quad \Delta = 1 \\ 1.3557 \pm 0.0092 \pm 0.0060 & \text{assuming} \quad \Delta = 11/16 \end{cases} \tag{5.9}$$

Table VII. Maximum-Likelihood Estimator $\hat{\gamma}$ As a Function of the Parameter $\tilde{a}_1$ and the Cut $N_{\min}$, for $\Delta=1$ and $\Delta=11/16$[a]

	$\tilde{a}_1$ $(\Delta=1)$									
$N_{\min}$	0.20	0.50	0.80	1.10	1.40	1.70	2.00	2.30	2.60	$\Delta\hat{\gamma}$
20	1.3464	1.3491	1.3517	1.3544	1.3570	1.3596	1.3621	1.3647	1.3673	0.0017
40	1.3472	1.3491	1.3510	1.3529	1.3547	1.3566	1.3585	1.3603	1.3621	0.0020
60	1.3476	1.3491	1.3506	1.3521	1.3536	1.3551	1.3566	1.3580	1.3595	0.0023
80	1.3473	**1.3486**	**1.3498**	**1.3511**	**1.3523**	**1.3536**	**1.3548**	**1.3560**	1.3573	0.0027
100	1.3482	**1.3493**	**1.3504**	**1.3515**	**1.3526**	**1.3536**	**1.3547**	**1.3558**	1.3568	0.0030
120	1.3491	**1.3500**	**1.3510**	**1.3519**	**1.3529**	**1.3539**	**1.3548**	**1.3558**	1.3567	0.0032
140	1.3494	**1.3503**	**1.3512**	**1.3521**	**1.3529**	**1.3538**	**1.3546**	**1.3555**	1.3564	0.0035
160	1.3490	**1.3498**	**1.3506**	**1.3514**	**1.3522**	**1.3529**	**1.3537**	**1.3545**	1.3553	0.0038
180	1.3493	**1.3501**	**1.3508**	**1.3516**	**1.3523**	**1.3530**	**1.3538**	**1.3545**	1.3552	0.0041
200	1.3499	**1.3506**	**1.3513**	**1.3520**	**1.3527**	**1.3534**	**1.3541**	**1.3547**	1.3554	0.0044

	$\tilde{a}_1$ $(\Delta=11/16)$									
$N_{\min}$	0.00	0.15	0.30	0.45	0.60	0.75	0.90	1.05	1.20	$\Delta\hat{\gamma}$
20	1.3446	1.3484	1.3521	1.3559	1.3595	1.3631	1.3667	1.3702	1.3737	0.0017
40	1.3460	1.3490	1.3520	1.3550	1.3580	1.3609	1.3638	1.3667	1.3696	0.0020
60	1.3466	1.3492	1.3518	1.3544	1.3570	1.3595	1.3621	1.3646	1.3671	0.0023
80	1.3465	1.3488	**1.3511**	**1.3535**	**1.3557**	**1.3580**	**1.3603**	1.3625	1.3647	0.0027
100	1.3475	1.3496	**1.3517**	**1.3538**	**1.3559**	**1.3580**	**1.3601**	1.3621	1.3642	0.0030
120	1.3484	1.3504	**1.3523**	**1.3543**	**1.3562**	**1.3582**	**1.3601**	1.3620	1.3639	0.0032
140	1.3489	1.3507	**1.3525**	**1.3544**	**1.3562**	**1.3580**	**1.3598**	1.3616	1.3633	0.0035
160	1.3484	1.3502	**1.3519**	**1.3536**	**1.3554**	**1.3571**	**1.3588**	1.3604	1.3621	0.0038
180	1.3489	1.3505	**1.3522**	**1.3538**	**1.3554**	**1.3571**	**1.3587**	1.3603	1.3619	0.0041
200	1.3495	1.3511	**1.3526**	**1.3542**	**1.3558**	**1.3573**	**1.3589**	1.3604	1.3619	0.0044

[a] Standard deviation $\Delta\hat{\gamma}$ is essentially independent of $\tilde{a}_1$.

where the first error bar represents systematic error due to corrections to scaling (95% subjective confidence limits) and the second error bar represents statistical error (95% classical confidence limits). We are unable to distinguish between the two proposed forms for the correction-to-scaling term. These estimates are, in any case, in good agreement with the believed exact value $\gamma = 43/32 = 1.34375$.[39,40]

We want to emphasize that these estimates of γ are based on an extremely modest CPU time (about 300 VAX hours; see Section 5.3 below); they are far from being the "state of the art" for estimating γ. Indeed, in this paper our main interest has been to analyze the dynamic critical behavior of the join-and-cut algorithm; we saw little value in investing large amounts of CPU time in order to estimate a critical exponent that is already known exactly! The reader should rest assured that we will make a more serious effort to estimate γ in the *three*-dimensional SAW.[46]

5.3. Computational Complexity

In Table VIII we give some information on the computational complexity of the algorithm. (Recall that we are using the "improved" algorithm described in Section 3.2.) The first column reports the probability of finding a "blank" hash table for the walk that is proposed to grow in length, conditioned on failure of the join-and-cut move.[16] The second and third columns report the mean number of hash-table insertions in a join-and-cut move, conditioned on either success or failure. The last column

[16] We also should have measured this quantity conditioned on *success*, but we forgot.

Table VIII. More Detailed Information on the Computational Complexity for the Two Types of Moves[a]

	Join-and-cut			Pivot
N_{tot}	Percent failures with blank hash table	Hash-table insertions per success	Hash-table insertions per failure	Hash-table insertions per failure
500	0.1363 ± 0.0004	200.1 ± 0.7	20.68 ± 0.05	20.63 ± 0.03
1000	0.1276 ± 0.0004	407.5 ± 1.7	32.06 ± 0.09	33.33 ± 0.04
2000	0.1165 ± 0.0004	827.8 ± 4.6	49.79 ± 0.18	54.76 ± 0.08
4000	0.1036 ± 0.0005	1683 ± 12	76.59 ± 0.36	90.23 ± 0.13
8000	0.0920 ± 0.0007	3434 ± 44	119.9 ± 0.9	151.0 ± 0.4

[a] Data are taken for $n_{\text{piv}} = 1$, but the last three columns are independent of n_{piv}.

reports the mean number of hash-table insertions in a pivot move, conditioned on failure. These data were obtained from the runs with $n_{\text{piv}} = 1$, but all except the first column are independent of n_{piv}.

In Section 3 we predicted that these columns should scale with critical exponents $1-\gamma+p \approx -0.15$, 1, $2-\gamma = 21/32 \approx 0.66$, and $1-p \approx 0.81$, respectively. Least-squares fits yield the following estimates:

Blank hash table:

$$\begin{cases} -0.133 \pm 0.002 \ (\chi^2 = 84.04, \text{ level} = 4 \times 10^{-18}) \\ \quad \text{using all data points} \\ -0.153 \pm 0.003 \ (\chi^2 = 14.52, \text{ level} = 0.07\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 1000 \\ -0.170 \pm 0.005 \ (\chi^2 = 0.012, \text{ level} = 91\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 2000 \end{cases} \tag{5.10}$$

Join-and-cut (success):

$$\begin{cases} 1.024 \pm 0.003 \ (\chi^2 = 0.10, \text{ level} = 99\,\%) \\ \quad \text{using all data points} \\ 1.024 \pm 0.005 \ (\chi^2 = 0.08, \text{ level} = 96\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 1000 \\ 1.025 \pm 0.009 \ (\chi^2 = 0.03, \text{ level} = 86\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 2000 \end{cases} \tag{5.11}$$

Join-and-cut (failure):

$$\begin{cases} 0.632 \pm 0.002 \ (\chi^2 = 2.04, \text{ level} = 56\,\%) \\ \quad \text{using all data points} \\ 0.632 \pm 0.003 \ (\chi^2 = 2.01, \text{ level} = 37\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 1000 \\ 0.630 \pm 0.005 \ (\chi^2 = 1.95, \text{ level} = 16\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 2000 \end{cases} \tag{5.12}$$

Pivot (failure):

$$\begin{cases} 0.715 \pm 0.001 \ (\chi^2 = 130.38, \text{ level} = 4 \times 10^{-28}) \\ \quad \text{using all data points} \\ 0.722 \pm 0.001 \ (\chi^2 = 27.29, \text{ level} = 10^{-6}) \\ \quad \text{using } N_{\text{tot}} \geqslant 1000 \\ 0.729 \pm 0.002 \ (\chi^2 = 13.79, \text{ level} = 0.02\,\%) \\ \quad \text{using } N_{\text{tot}} \geqslant 2000 \end{cases} \tag{5.13}$$

Table IX. CPU Time per Iteration, in Milliseconds, on a VAX 8650 Running VMS Fortran Version 5.6[a]

N_{tot}	$n_{\text{piv}}=1$	$=2$	$=5$	$=10$	$=20$	$=40$	$=80$	$=160$
500	10	17	38	74	146	289	577	1149
1000	16	28	62	123	243	477	953	
3000	27	47	105	206	406	811		
4000	44	78	181	352	703	1381		
8000	68	135	309	596	1194	2395		

[a] Errors are of the order of 1%.

The data in the first and fourth columns show strong curvature on a log-log plot, as reflected by the poor goodness of fit. Hence these exponent estimates are presumably afflicted by a large systematic error. In any case, all but the last column are in reasonably good agreement with the predicted exponents.

Our measurements of hash-table insertions in the pivot algorithm agree perfectly with those of Madras and Sokal (ref. 3, Section 4.4, Table VI), if one makes the correspondence $N \leftrightarrow \langle N \rangle = N_{\text{tot}}/2$.[17] Indeed, Madras and Sokal found an exponent 0.745 ± 0.005, in good agreement with ours, and were puzzled by the discrepancy with the predicted exponent $1-p \approx 0.81$. However, our reanalysis of the Madras–Sokal data shows that the exponent drifts upward from 0.745 to 0.777 as the data points at low N are successively removed, with every indication that the upward trend would persist for $N > 10000$. Our own data from the last column of Table VIII show a similar but weaker trend. Therefore, the discrepancy from the predicted exponent can plausibly be explained as an effect of corrections to scaling. In any case, it would be useful to clarify this issue in future work on the pivot algorithm.

In Table IX we report the CPU time per iteration (on a VAX 8650 running VMS Fortran version 5.6) for the join-and-cut algorithm, as a function of N_{tot} and n_{piv}. These measurements fit reasonably well the form

$$T_{\text{CPU}} = aN_{\text{tot}}^{2-\gamma} + bn_{\text{piv}}N_{\text{tot}}^{1-p} \tag{5.14}$$

predicted in (3.12), with $a \approx 0.045$ and $b \approx 0.043$.

[17] In our algorithm, N ($=N_1$ or N_2) is a random variable with the distribution (4.1). However, since $E(\text{work}) \sim N^{1-p}$ with $1-p$ close to 1, we have $\langle E(\text{work}) \rangle \sim \langle N^{1-p} \rangle \approx \langle N \rangle^{1-p}$. Hence it is reasonable to compare our data (N variable with those of Madras and Sokal (N fixed).

Our simulations took a total of approximately 1600 hr of CPU time on a VAX 8650 running VMS Fortran. Most of this time was spent in the simulations at large n_{piv}, which were needed for our study of the dynamic critical behavior. Only about 230 hr was spent at $n_{\text{piv}} = 1$, which contributed the bulk of the statistics used in the determination of γ.

6. CONCLUSIONS

In this paper we have introduced a new algorithm for simulating self-avoiding walks with variable length and free endpoints, and thereby estimating the critical exponent γ. We have analyzed rigorously an "idealized" version of the algorithm (Sections 2.1–2.3), and have proposed heuristic scaling relations for the dynamic critical exponents that arise in the "practical" version of the algorithm (Sections 2.4–2.5). In two dimensions, we have measured these exponents, and have verified the scaling relations to modest accuracy (Section 5.1).

The best choice in practice is $n_{\text{piv}} = 1$: empirically this minimizes the autocorrelation time $\tau_{\text{int},X}$ measured in CPU units, as can easily be seen by multiplying the entries in Tables VI and IX. For $n_{\text{piv}} = 1$ (or any *fixed* n_{piv}), the autocorrelation time $\tau_{\text{int},X}$ measured in units of *iterations* scales as

$$\tau_{\text{int},X} \sim N_{\text{tot}}^{q_X} \tag{6.1}$$

where

$$\begin{aligned} q_X &= r_X + a_X s_X \\ &= \gamma - 1 + a_X s_X \end{aligned} \tag{6.2}$$

Here the exponents r_X and s_X are defined in the scaling Ansatz (2.35), and a_X is defined in (2.45). In (2.44) we predict $s_X \approx p + \gamma - 1$ (where p is the acceptance-fraction exponent of the pivot algorithm), but we are unable to predict a_X. On the other hand, the CPU time per iteration scales as

$$E(\text{work per iteration}) \sim N_{\text{tot}}^{1-p} \tag{6.3}$$

[see (3.13)]. Therefore, the autocorrelation time measured in CPU units scales as

$$\tau_{\text{int},X}^{(\text{CPU})} \sim N_{\text{tot}}^{k} \tag{6.4}$$

where

$$\begin{aligned} k &= q_X + (1 - p) \\ &= \gamma - p + a_X s_X \end{aligned} \tag{6.5}$$

The exponent k thus controls the statistical efficiency of the algorithm.

In dimension $d = 2$, our measurements yield

$$q_X \approx 0.70 \pm 0.05 \tag{6.6}$$

$$1 - p \approx 0.81 \pm 0.01 \tag{6.7}$$

and hence

$$k \approx 1.51 \pm 0.06 \tag{6.8}$$

This is a significant improvement over the behavior $\tau_{\mathrm{int}}^{(\mathrm{CPU})} \sim N^{\approx 2}$ of the Berretti–Sokal algorithm.[9-11]

In dimensions $d > 2$ we expect that the behavior of the join-and-cut algorithm will improve further. Indeed, it is known that

$$r_X = \gamma - 1 = \begin{cases} 0.34375 & \text{for} \quad d = 2^{(39, 40)} \\ \approx 0.16 & \text{for} \quad d = 3^{(41)} \\ 0 & \text{for} \quad d \geqslant 4 \end{cases} \tag{6.9}$$

and

$$1 - p \approx \begin{cases} 0.81 & \text{for} \quad d = 2^{(3)} \\ 0.89 & \text{for} \quad d = 3^{(27)} \\ 1 & \text{for} \quad d \geqslant 4 \quad \text{(predicted)} \end{cases} \tag{6.10}$$

—in both cases modulo logarithms for $d = 4$. [Here the prediction for p in $d \geqslant 4$ follows from the empirical inequality $0 \leqslant p \leqslant \gamma - 1$ as well as from the heuristic arguments (ref. 3, Section 3.2) relating p to $\gamma - 1$.] From this we deduce that

$$s_X \approx p + \gamma - 1 \approx \begin{cases} 0.54 & \text{for} \quad d = 2 \\ 0.27 & \text{for} \quad d = 3 \\ 0 & \text{for} \quad d \geqslant 4 \end{cases} \tag{6.11}$$

Unfortunately we have no precise prediction for the exponent a_X. But provided only that a_X remains *finite* for $d \geqslant 4$, we can conclude that $a_X s_X = 0$ for $d \geqslant 4$ and hence that

$$k = 1 \qquad \text{for} \quad d \geqslant 4 \tag{6.12}$$

(modulo logarithms in $d = 4$). If, in addition, the exponent k varies in a reasonably sooth and monotonic manner as a function of d, then it is reasonable to guess that $k \approx 1.2$–1.3 in $d = 3$.

Thus, in dimension $d \geqslant 4$ we expect the join-and-cut algorithm to generate an "effectively independent" sample from the distribution (1.3) in a CPU time of order N_{tot}. This behavior is essentially optimal, since it takes a time of order N merely *to write down* an N-step walk![18] Even in dimensions $d=2, 3$ the behavior is not far from optimal.

The behavior $\tau_{\text{int},X}^{(\text{CPU})} \sim N_{\text{tot}}^{k}$ of the join-and-cut algorithm can be compared with the behavior $\tau_{\text{int}}^{(\text{CPU})} \sim N$ of the pivot algorithm. In a certain sense, the residual factor N^{k-1} is the price we pay (in dimensions $d<4$) for obtaining an ensemble of SAWs with *variable* rather than fixed N.

Using the join-and-cut algorithm, we expect to produce in the near future a high-precision Monte Carlo determination of the exponent γ in the $d=3$ SAW.(46) In dimension $d=4$ we would like to measure the logarithmic exponent $\bar{\gamma}$ defined by

$$c_N \sim \mu^N (\log N)^{\bar{\gamma}} \tag{6.13}$$

and in particular to test the renormalization-group prediction(47-49) $\bar{\gamma}=1/4$. From a statistical point of view this problem is several times more difficult than estimating the leading exponent γ, because var(log log N) is several times smaller than var(log N). However, the most serious difficulty may come from corrections to scaling, which are expected to be of the form

$$\begin{aligned} c_N \sim \mu^N (\log N)^{\bar{\gamma}} a_0 \Bigg[& 1 + b_{11} \frac{\log \log N}{\log N} + b_{01} \frac{1}{\log N} \\ & + b_{12} \frac{\log \log N}{(\log N)^2} + b_{02} \frac{1}{(\log N)^2} + \cdots \Bigg] \end{aligned} \tag{6.14}$$

where b_{11} is a universal coefficient that can be computed from the renormalization group, and the other coefficients are nonuniversal. The corrections to scaling are thus *extremely* slowly decreasing in N. The extraordinary efficiency of the join-and-cut algorithm will allow us to go to very large values of N, of order 10000 or more. But it remains to be seen whether such values of N are large enough to see clearly the true asymptotic behavior.

APPENDIX

More rigorous statistical analysis shows that the fit to a straight line in Fig. 3 is not as good as it seems: for $s_X = 0.54$, the χ^2 values are

[18] In principle, one could imagine generating directly the correct probability distribution of N_1, without actually generating the walks (ω_1, ω_2). But we cannot conceive of how to implement such a scheme in practice, short of solving analytically for the $\{c_N\}$.

$$\chi^2 = \begin{cases} 101.60 & (31 \text{ d.f., level} = 2 \times 10^{-9}) & \text{using all data points} \\ 60.89 & (23 \text{ d.f., level} = 0.003\,\%) & \text{using } N_{\text{tot}} \geqslant 1000 \\ 28.97 & (16 \text{ d.f., level} = 2.4\,\%) & \text{using } N_{\text{tot}} \geqslant 2000 \\ 18.99 & (10 \text{ d.f., level} = 4.0\,\%) & \text{using } N_{\text{tot}} \geqslant 4000 \\ 9.17 & (4 \text{ d.f., level} = 5.7\,\%) & \text{using } N_{\text{tot}} = 8000 \end{cases}$$

(For $s_X = 0.63$ the χ^2 values are slightly larger.) However, these values of χ^2 could be explained if for some reason we had underestimated the error bars on $\tau_{\text{int},X}$ by a factor of ≈ 1.4. The straight lines in Fig. 3 are the weighted-least-squares fits to the data with $N_{\text{tot}} \geqslant 2000$; they are given by

$$F_X(z) = \begin{cases} 0.20275 + 0.00440/z & \text{for} \quad s_X = 0.54 \\ 0.20370 + 0.00207/z & \text{for} \quad s_X = 0.63 \end{cases}$$

ACKNOWLEDGMENTS

One of us (A.D.S.) wishes to thank Neal Madras for valuable conversations in which the ideas of a multi-SAW ensemble and of concatenation-type moves were first explored.

This research was supported in part by the Istituto Nazionale di Fisica Nucleare (S.C. and A.P.), U.S. National Science Foundation grant DMS-8911273 (A.D.S.), U.S. Department of Energy contract DE-FG02-90ER40581 (A.D.S.), and NATO Collaborative Research Grant CRG 910251 (S.C. and A.D.S.). Acknowledgment is also made to the donors of the Petroleum Research Fund, administered by the American Chemical Society, for partial support of this research.

REFERENCES

1. M. Lal, *Molec. Phys.* **17**:57 (1969).
2. B. MacDonald, N. Jan, D. L. Hunter, and M. O. Steinitz, *J. Phys. A* **18**:2627 (1985).
3. N. Madras and A. D. Sokal, *J. Stat. Phys.* **50**:109 (1988).
4. A. T. Clark and M. Lal, *Br. Polymer J.* **9**:92 (1977).
5. S. Caracciolo, A. Pelissetto, and A. D. Sokal, *J. Stat. Phys.* **60**:1 (1990).
6. L. E. Dubins, A. Orlitsky, J. A. Reeds, and L. A. Shepp, *IEEE Trans. Inform. Theory* **34**:1509 (1988).
7. N. Madras, A. Orlitsky, and L. A. Shepp, *J. Stat. Phys.* **58**:159 (1990).
8. E. J. Janse van Rensburg, S. G. Whittington, and N. Madras, *J. Phys. A* **23**:1589 (1990).
9. A. Berretti and A. D. Sokal, *J. Stat. Phys.* **40**:483 (1985).
10. G. F. Lawler and A. D. Sokal, *Trans. Am. Math. Soc.* **309**:557 (1988).
11. A. D. Sokal and L. E. Thomas, *J. Stat. Phys.* **54**:797 (1989).
12. S. Redner and P. J. Reynolds, *J. Phys. A* **14**:2679 (1981).
13. B. Berg and D. Foerster, *Phys. Lett.* **106B**:323 (1981).

14. C. Aragão de Carvalho, S. Caracciolo, and J. Fröhlich, *Nucl. Phys. B* **215**[FS7]:209 (1983).
15. C. Aragão de Carvalho and S. Caracciolo, *J. Phys.* (Paris) **44**:323 (1983).
16. A. D. Sokal and L. E. Thomas, *J. Stat. Phys.* **51**:907 (1988).
17. S. Caracciolo, A. Pelissetto, and A. D. Sokal, *J. Stat. Phys.* **63**:857 (1991).
18. S. Caracciolo, A. Pelissetto, and A. D. Sokal, *Nucl. Phys. B (Proc. Suppl.)* **20**:68 (1991).
19. S. D. Silvey, *Statistical Inference* (Chapman and Hall, London, 1975), Chapter 4.
20. A. D. Sokal, Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms, Cours de Troisième Cycle de la Physique en Suisse Romande (Lausanne, June 1989).
21. R. G. Edwards and A. D. Sokal, *Phys. Rev. D* **38**:2009 (1988).
22. S. Caracciolo, R. G. Edwards, A. Pelissetto, and A. D. Sokal, *Nucl. Phys. B (Proc. Suppl.)* **20**:72 (1991); and paper in preparation.
23. P. R. Halmos, *A Hilbert Space Problem Book*, 2nd ed. (Springer, New York, 1982).
24. T. Kato, *Perturbation Theory for Linear Operators*, 2nd ed. (Springer, New York, 1976), Section I.6.10.
25. J. Keilson, *Markov Chain Models—Rarity and Exponentiality* (Springer, New York, 1979).
26. P. Lancaster, *Theory of Matrices* (Academic Press, New York, 1969), Chapter 3.
27. S. Caracciolo, G. Ferraro, and A. Pelissetto, in preparation.
28. D. E. Knuth, *The Art of Computer Programming*, Vol. 3 (Addison-Wesley, Reading, Massachusetts, 1973), Section 6.4.
29. S. S. Wilks, *Mathematical Statistics* (Wiley, New York, 1962).
30. P. J. Huber, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1 (University of California Press, Berkeley, 1967), pp. 221–233.
31. I. V. Basawa and B. L. S. Prakasa Rao, *Statistical Inference for Stochastic Processes* (Academic Press, London, 1980), pp. 127–134, 162, 164.
32. Y. Ogata, *J. Appl. Prob.* **17**:59 (1980).
33. A. Azzalini, *Biometrika* **70**:381 (1983).
34. P. Billingsley, *Convergence of Probability Measures* (Wiley, New York, 1968).
35. R. Cogburn, in *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 2 (University of California Press, Berkeley, 1972), pp. 485–512.
36. S. Niemi and E. Nummelin, *Commentationes Physico-Mathematicae*, No. 54 (Societas Scientiarum Fennicae, Helsinki, 1982).
37. D. Landers and L. Rogge, *Z. Wahrsch. Verw. Gebiete* **35**:57 (1976).
38. E. Nummelin, *General Irreducible Markov Chains and Non-Negative Operators* (Cambridge University Press, Cambridge, 1984).
39. B. Nienhuis, *Phys. Rev. Lett.* **49**:1062 (1982).
40. B. Nienhuis, *J. Stat. Phys.* **34**:731 (1984).
41. A. J. Guttmann, *J. Phys. A* **22**:2807 (1989).
42. F. J. Wegner, *Phys. Rev. B* **5**:4529 (1972).
43. A. J. Guttmann, T. R. Osborn, and A. D. Sokal, *J. Phys. A* **19**:2591 (1986).
44. H. Saleur, *J. Phys. A* **20**:455 (1987).
45. S. Caracciolo, G. Ferraro, and A. Pelissetto, *J. Phys. A* **24**:3625 (1991).
46. S. Caracciolo, A. Pelissetto, and A. D. Sokal, in preparation.
47. A. I. Larkin and D. E. Khmel'nitskii, *Zh. Eksp. Teor. Fiz.* **56**:2087 (1969) [*Sov. Phys.-JETP* **29**:1123 (1969)].
48. F. J. Wegner and E. K. Riedel, *Phys. Rev. B* **7**:248 (1973).
49. E. Brézin, J.-C. LeGuillou, and J. Zinn-Justin, *Phys. Rev. D* **8**:2418 (1973).