Journal of Statistical Physics, Vol. 50, Nos. 1/2, 1988

The Pivot Algorithm: A Highly Efficient Monte Carlo Method for the Self-Avoiding Walk

Neal Madras[1] and Alan D. Sokal[2]

Received June 19, 1987

The pivot algorithm is a dynamic Monte Carlo algorithm, first invented by Lal, which generates self-avoiding walks (SAWs) in a canonical (fixed-N) ensemble with free endpoints (here N is the number of steps in the walk). We find that the pivot algorithm is extraordinarily efficient: one "effectively independent" sample can be produced in a computer time of order N. This paper is a comprehensive study of the pivot algorithm, including: a heuristic and numerical analysis of the acceptance fraction and autocorrelation time; an exact analysis of the pivot algorithm for ordinary random walk; a discussion of data structures and computational complexity; a rigorous proof of ergodicity; and numerical results on self-avoiding walks in two and three dimensions. Our estimates for critical exponents are $\nu = 0.7496 \pm 0.0007$ in $d = 2$ and $\nu = 0.592 \pm 0.003$ in $d = 3$ (95% confidence limits), based on SAWs of lengths $200 \leqslant N \leqslant 10000$ and $200 \leqslant N \leqslant 3000$, respectively.

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

1. INTRODUCTION

The self-avoiding walk (SAW) is a well-known lattice model of a polymer molecule with excluded volume.[1-3] Its equivalence to the $N = 0$ limit of the N-vector model[4-9] has also made it an important test case in the theory of critical phenomena.

Monte Carlo studies of the SAW go back to the early 1950s,[10] and in recent years several improved Monte Carlo algorithms for the SAW have been devised.[11-13] In this paper we study yet another algorithm, which, though not new, turns out to be extraordinarily efficient. This algorithm, which we call the *pivot algorithm*, was invented in 1969 by Lal,[14] used

[1] Department of Mathematics, University of Toronto, Toronto, Ontario, Canada M5S 1A1.

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

in the mid-1970s by Olaj and Pelinka,[15] and reinvented in 1985 by MacDonald *et al.*[16,17] Continuum analogues of the pivot algorithm have been used by Stellman and Gans[18,19] and Freire and Horta.[20] Except for these few references, however, the pivot algorithm seems to have rested in oblivion. This is a shame, for, as we shall demonstrate, the pivot algorithm appears to be the *most efficient algorithm yet invented* for estimating the critical exponent ν in the self-avoiding walk.

The pivot algorithm is a *dynamic* Monte Carlo algorithm, which generates SAWs in a *canonical ensemble* (fixed number of steps N) with *free endpoints*. The elementary move of the algorithm is as follows: A site on the walk is chosen at random and used as a pivot point; a random symmetry operation of the lattice (e.g., rotation or reflection) is applied to the part of the walk subsequent to the pivot point, using the pivot point as the origin. The resulting walk is accepted if it is self-avoiding; otherwise, it is rejected, and the old walk is counted once again in the sample. It is not hard to prove that this algorithm is ergodic (see Section 3.5) and satisfies detailed balance for the standard equal-weight SAW probability distribution.

At first thought this seems to be a terrible algorithm: for N large, nearly all the proposed moves will get rejected. In fact, this latter statement is true, but the hasty conclusion drawn from it is radically false! The acceptance fraction f does indeed go to zero as $N \to \infty$, roughly like N^{-p}; numerically, we find that in two dimensions the exponent p is around 0.19. But this means that roughly once every N^p moves one gets an acceptance. And the pivot moves are very radical: one might surmise that after very few accepted moves (say, five or ten) the SAW will have reached an "essentially new" configuration. One conjectures, therefore, that the autocorrelation time τ of the pivot algorithm behaves as $\sim N^p$. Things are in fact somewhat more subtle (see Sections 3.2 and 3.3), but roughly speaking (and modulo a possible logarithm) this conjecture appears to be true. On the other hand, a careful analysis of the computational complexity of the pivot algorithm (Section 3.4) shows that one accepted move can be produced in a computer time of order N. Combining these two facts, we conclude that one "effectively independent" sample (at least as regards global observables) can be produced in a computer time of order N (or perhaps $N \log N$). This is a factor $\sim N$ more efficient than the alternative algorithms due to Redner and Reynolds[11] and Berretti and Sokal.[13] Indeed, this order of efficiency cannot be surpassed by any algorithm that computes each site on successive SAWs, for it takes a time of order N simply to write down an N-step walk![3]

[3] A personal aside: One of the authors (A.D.S.) was familiar with the Lal paper as early as the fall of 1982, but rejected the algorithm out of hand on the ground that the acceptance

The plan of this paper is as follows: In Section 2 we give a brief review of the self-avoiding walk (SAW) and dynamic Monte Carlo methods, and set the notation. The heart of the paper is Section 3: in successive subsections we define the pivot algorithm and some of its variants; give a heuristic analysis of its acceptance fraction and autocorrelation time; carry out an exact analysis of the pivot algorithm for the ordinary random walk; discuss the data structures needed and analyze the computational complexity; prove rigorously the ergodicity of the algorithm; and discuss questions relating to initialization. In Section 4 we present numerical results using the pivot algorithm on two-dimensional SAW's of lengths $200 \leqslant N \leqslant 10,000$. (This latter length is probably a world record, if anyone cares.) We also present preliminary results on three-dimensional SAWs of lengths $200 \leqslant N \leqslant 3000$. These simulations used a total of roughly 300 and 120 hr CPU time, respectively, on a Cyber 170-730 computer. Our estimates for ν are 0.7496 ± 0.0007 in $d=2$, and 0.592 ± 0.003 in $d=3$ (95% confidence limits). In Section 5 we discuss prospects for the future, including a test of the hyperscaling relation $d\nu = 2\Delta_4 - \gamma$ in dimension $d=3$ and a search for logarithmic violation of hyperscaling in dimension $d=4$. In Appendix A we present an exact-enumeration/series-extrapolation analysis of the acceptance fraction. In Appendix B we prove some bounds on the eigenvalues of the pivot algorithm for ordinary random walk. In Appendix C we discuss our statistical methods.

2. BACKGROUND AND NOTATION

2.1. The Self-Avoiding Walk (SAW): A Review

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

fraction would be tiny for large N. In June 1984 A.D.S. met Freire in Madrid, and made the same argument to him. In May 1985, the other author (N.M.) proposed the pivot algorithm to A.D.S. in discussion after a lecture on Monte Carlo methods for the SAW. The response of A.D.S. was "Oh, yes, I know that algorithm, it's a terrible algorithm because... ." N.M. forced A.D.S. to take a second look, and within days we convinced ourselves heuristically that the acceptance fraction would go to zero only as a very weak power law (see Section 3.2). We then started work on this paper in earnest. A.D.S. wants to take this opportunity to atone publicly for his sins, and in particular to apologize to Juan Freire for having criticized his work unjustly.

attention to the simple (hyper)cubic lattice $\mathbb{Z}^d$; similar ideas would apply to other regular lattices. We assume all walks to begin at the origin ($\omega_0 = 0$) unless stated otherwise.

Let $\mathcal{S}_N$ [respectively, $\mathcal{S}_N(x)$] be the set of N-step SAWs on $\mathbb{Z}^d$ starting at the origin and ending anywhere [respectively, ending at x]; and let c_N [respectively, $c_N(x)$] be the cardinality of $\mathcal{S}_N$ [respectively, of $\mathcal{S}_N(x)$]. Then it can be proven[21,22] that the limits

$$\mu = \lim_{N \to \infty} c_N^{1/N} = \lim_{\substack{N \to \infty \\ N = x \bmod 2}} c_N(x)^{1/N} \qquad (x \text{ fixed} \neq 0) \tag{2.1}$$

exist and are equal. Here μ is a positive constant called the *connective constant* (or *effective coordination number*) of the lattice. Some slightly stronger bounds on c_N and $c_N(x)$ can also be proven.[23-25] It is *believed*, but not yet proven, that c_N and $c_N(x)$ have the asymptotic behavior

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

$$c_N(x) \sim \mu^N N^{\alpha_{\text{sing}} - 2} \qquad (x \text{ fixed} \neq 0) \tag{2.3}$$

as $N \to \infty$. Here γ and α_{sing} are *critical exponents*, which are believed to be universal among lattices of a given dimension d.

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

$$\langle \omega_N^2 \rangle \equiv \frac{1}{c_N} \sum_x |x|^2 \, c_N(x) \tag{2.4}$$

and the mean-square radius of gyration

$$\langle S_N^2 \rangle \equiv \frac{1}{c_N} \sum_{\omega \in \mathcal{S}_N} S_N^2(\omega) \tag{2.5}$$

where

$$S_N^2(\omega) \equiv \frac{1}{N+1} \sum_{i=0}^{N} \left(\omega_i - \frac{1}{N+1} \sum_{j=0}^{N} \omega_j \right)^2 \tag{2.6a}$$

$$= \frac{1}{N+1} \sum_{i=0}^{N} \omega_i^2 - \left(\frac{1}{N+1} \sum_{i=0}^{N} \omega_i \right)^2 \tag{2.6b}$$

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

$$\langle \omega_N^2 \rangle \sim N^{2\nu} \tag{2.7}$$

$$\langle S_N^2 \rangle \sim N^{2\nu} \tag{2.8}$$

as $N \to \infty$, where ν is another (universal) critical exponent. [Very recently, Slade[26] has proven that (2.7) holds with $\nu = 1/2$ for SAWs in sufficiently high dimension d.]

Finally, let c_{N_1, N_2} be the number of pairs $(\omega^{(1)}, \omega^{(2)})$ such that $\omega^{(1)}$ is an N_1-step SAW starting at the origin, $\omega^{(2)}$ is an N_2-step SAW starting *anywhere*, and $\omega^{(1)}$ and $\omega^{(2)}$ have at least one point in common (i.e., $\omega^{(1)} \cap \omega^{(2)} \neq \varnothing$). (This quantity is closely related to the osmotic second virial coefficient for polymer molecules.) Then it is believed that

$$c_{N_1, N_2} \sim \mu^{N_1 + N_2} N_1^{2\Delta_4 + \gamma - 2} g(N_1/N_2) \tag{2.9}$$

as N_1, $N_2 \to \infty$, where Δ_4 is yet another (universal) critical exponent and g is a (universal) scaling function.

2.2. Dynamic Monte Carlo Methods: A Review

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

Monte Carlo methods can be classified as *static* or *dynamic*. Static methods are those that generate a sequence of *statistically independent* samples from the desired probability distribution π. Dynamic methods are those that generate a sequence of *correlated* samples from some stochastic process (usually a Markov process) having the desired probability distribution π as its unique equilibrium distribution.

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

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

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

$$\sum_{i \in S} \pi_i p_{ij} = \pi_j \tag{2.10}$$

(This condition asserts that π is a stationary distribution for the Markov chain $P = \{p_{ij}\}$.)

In this case it can be shown[27-29] that π is the *unique* stationary distribution for the Markov chain $P = \{p_{ij}\}$, and that the occupation-time distribution over long time intervals converges (with probability 1) to π, irrespective of the initial state of the system. If, in addition, P is *aperiodic* (this means that for each pair $i, j \in S$, $p_{ij}^{(n)} > 0$ for *all* sufficiently large n), then the probability distribution at any single time in the far future also converges to π, irrespective of the initial state, that is, $\lim_{n \to \infty} p_{ij}^{(n)} = \pi_j$ for all i.

Thus, simulation of the Markov chain P provides a legitimate Monte Carlo method for estimating averages with respect to π. However, since the successive states $X_0, X_1,...$ of the Markov chain are in general highly correlated, the variance of estimates produced in this way may be much higher than in independent sampling. To make this precise, let $A = \{A(i)\}_{i \in S}$ be a real-valued function defined on the state space S (i.e., a real-valued observable); and consider the *stationary* Markov chain (i.e., start the system in the stationary distribution π, or equivalently, "thermalize" it for a very long time prior to observing the system). Then $\{A_t\} \equiv \{A(X_t)\}$ is a stationary stochastic process with mean

$$\mu_A \equiv \langle A_t \rangle = \sum_{i \in S} \pi_i A(i) \tag{2.11}$$

and *unnormalized autocorrelation function*[4]

$$\begin{aligned} C_{AA}(t) &\equiv \langle A_s A_{s+t} \rangle - \mu_A^2 \\ &= \sum_{i,j \in S} A(i)[\pi_i p_{ij}^{(|t|)} - \pi_i \pi_j]\, A(j) \end{aligned} \tag{2.12}$$

The *normalized autocorrelation function* is then

$$\rho_{AA}(t) \equiv C_{AA}(t)/C_{AA}(0) \tag{2.13}$$

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

$$\tau_{\exp, A} = \limsup_{t \to \infty} \frac{t}{-\log |\rho_{AA}(t)|} \tag{2.14}$$

and

$$\tau_{\exp} = \sup_A \tau_{\exp, A} \tag{2.15}$$

Thus, $\tau_{\exp}$ is the relaxation time of the slowest mode in the system. (If the state space is infinite, $\tau_{\exp}$ might be $+\infty$. However, for an irreducible, aperiodic Markov chain on a finite state space, $\tau_{\exp}$ is always finite.)

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

An equivalent definition, which is useful for rigorous analysis, involves considering the eigenvalues of the transition probability matrix P. By the Perron–Frobenius theorem,[30] P has a nondegenerate eigenvalue 1 with right eigenvector equal to the constant function and left eigenvector equal to π; if P is aperiodic, then this is the only eigenvalue on the unit circle; and the remainder of the spectrum lies in the interior of the unit circle.[5] Let R be the spectral radius of the remainder of P, i.e.,

$$R \equiv \inf\{r\colon \operatorname{spec} P \subset \{\lambda\colon |\lambda| \leqslant r\} \cup \{1\}\} \tag{2.16}$$

Then, it is not difficult to show that $R = \exp(-1/\tau_{\mathrm{exp}})$. In particular, the rate of convergence to equilibrium from an initial nonequilibrium distribution is controlled by R, and hence by τ_{exp}.

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

$$\tau_{\mathrm{int},A} = \frac{1}{2} \sum_{t=-\infty}^{\infty} \rho_{AA}(t) = \frac{1}{2} + \sum_{t=1}^{\infty} \rho_{AA}(t) \tag{2.17}$$

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

$$\bar{A} \equiv \frac{1}{n} \sum_{t=1}^{n} A_t \tag{2.18}$$

has variance

$$\begin{aligned} \operatorname{var}(\bar{A}) &= \frac{1}{n^2} \sum_{r,s=1}^{n} C_{AA}(r-s) \\ &= \frac{1}{n} \sum_{t=-(n-1)}^{n-1} \left(1 - \frac{|t|}{n}\right) C_{AA}(t) && (2.19a) \\ &\approx \frac{1}{n} (2\tau_{\mathrm{int},A})\, C_{AA}(0) \qquad \text{for} \quad n \gg \tau && (2.19b) \end{aligned}$$

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

[5] In fact, P is a *contraction* with respect to any of the $l^p(\pi)$ norms $(1 \leqslant p \leqslant \infty)$. These norms are defined by $\|A\|_p \equiv (\sum_{i\in S} \pi_i\, |A(i)|^p)^{1/p}$ for $p < \infty$, and $\|A\|_\infty \equiv \sup_{i \in S} |A(i)|$. The $l^2(\pi)$ norm is particularly useful, as it arises from an inner product $(A, B) \equiv \sum_{i\in S} \pi_i A(i)^* B(i)$.

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

Most commonly, τ_{exp} and $\tau_{\text{int},A}$ are of the same order of magnitude, at least for "reasonable" observables A. In this case, the problem of initialization bias, i.e., the need to discard data at the beginning of a run, is not a serious one: perhaps $20\tau_{\text{exp}}$ data points at the beginning of the run need to be discarded in order to avoid a severe *systematic* error, but the total run length will have be on the order of $1000\tau_{\text{int},A}$ (or even longer) in order to obtain reasonably small *statistical* errors; so only a negligible fraction of the data is being discarded. However, some algorithms, such as the one studied in this paper, have the property that $\tau_{\text{exp}} \gg \tau_{\text{int},A}$ for the observables A of interest. In such cases the problem of initialization bias is potentially a serious one: the need to discard data at the beginning of the run could seriously degrade the computational efficiency of the algorithm. We return to this question in Section 3.6.

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

$$\text{(B}'\text{)} \quad \text{For each pair } i, j \in S, \ \pi_i p_{ij} = \pi_j p_{ji} \tag{2.20}$$

[Summing (B′) over i, we recover (B).] (B′) is called the *detailed-balance condition*; a Markov chain satisfying (B′) is called *reversible.*[6] Condition (B′) is equivalent to the *self-adjointness* of P as on operator on the space $l^2(\pi)$ (see footnote 4, above). In this case, it follows from the spectral theorem that the autocorrelation function $C_{AA}(t)$ has a spectral representation

$$C_{AA}(t) = \sum_{k=2}^{|S|} \alpha_{AA}^{(k)} \lambda_k^{|t|} \tag{2.21}$$

where $1 \equiv \lambda_1 > \lambda_2 \geqslant \lambda_3 \geqslant \cdots \geqslant \lambda_{|S|} \geqslant -1$ are the eigenvalues of P, and the spectral weights $\alpha_{AA}^{(k)}$ are *nonnegative.* Moreover, we have

$$R = \exp(-1/\tau_{\text{exp}}) = \max(|\lambda_2|, |\lambda_{|S|}|) \tag{2.22}$$

and

$$\tau_{\text{int},A} \leqslant \frac{1}{2}\frac{1+\lambda_2}{1-\lambda_2} \leqslant \frac{1}{2}\frac{1+\exp(-1/\tau_{\text{exp}})}{1-\exp(-1/\tau_{\text{exp}})} \approx \tau_{\text{exp}} \tag{2.23}$$

[6] For the physical significance of this term, see Kemeny and Snell (Ref. 27, Section 5.3) or Iosifescu (Ref. 28, Section 4.5).

3. THE PIVOT ALGORITHM

3.1. Definition of the Algorithm

The pivot algorithm[7] is a *dynamic* Monte Carlo algorithm, which generates SAWs in a *canonical ensemble* (fixed number of steps N) with one endpoint fixed at the origin ($\omega_0 = 0$) and the other endpoint *free*. The state space is thus $\mathscr{S}_N$, and the invariant probability measure is the standard equal-weight SAW distribution ($\pi_\omega = 1/c_N$ for each $\omega \in \mathscr{S}_N$). The elementary move of the pivot algorithm is as follows: choose at random a pivot point k along the walk ($0 \leqslant k \leqslant N-1$); choose at random an element of the symmetry group of the lattice (rotation or reflection or a combination thereof); then apply the symmetry-group element to $\omega_{k+1},..., \omega_N$ using ω_k as a pivot (i.e., as the temporary "origin"). The resulting walk is accepted if it is self-avoiding; otherwise it is rejected and the walk ω is counted once more in the sample.

Different variants of the pivot algorithm are obtained by specifying different distributions when we "choose at random":

1. The pivot point k can be chosen according to any preset family of strictly positive probabilities $p_0, p_1,..., p_{N-1}$. The strict positivity ($p_k > 0$ for all k) is necessary to ensure the ergodicity of the algorithm. In this paper we consider primarily a uniform distribution ($p_k = 1/N$ for all k); but see Section 5.2 for a discussion of situations in which other choices may possibly be advantageous.

[In practice we need not use $k = 0$ as a pivot point, since global rotations or reflections of the walk can be incorporated implicitly in the data analysis rather than explicitly in the simulation. Thus, the Markov chain that we in fact simulate is *not* ergodic on the whole space $\mathscr{S}_N$, but only on a *subset* of $\mathscr{S}_N$ consisting of walks whose first step is in a specified direction. However, each walk produced in the simulation is considered during the data analysis to be a "proxy" for itself *and all walks equivalent to it by global symmetries.* Actually, this latter step occurs automatically, since all the observables that we study (e.g., ω_N^2) are invariant under the symmetry group of the lattice. The simulations reported in this paper employ a uniform distribution: $p_k = 1/(N-1)$ for all $1 \leqslant k \leqslant N-1$.]

2. Let G be the group of orthogonal transformations (about the origin) that leave invariant the lattice $\mathbb{Z}^d$. Then the symmetry operation $g \in G$ can be chosen according to any preset probability distribution

[7] MacDonald *et al.*[(16,17)] call this the "wiggle" algorithm. We feel, however, that the term "pivot" more accurately describes the elementary move of the algorithm, and in particular emphasizes its global (nonlocal) nature.

$\{p_g\}_{g\in G}$ that satisfies $p_g = p_{g^{-1}}$ for all g, and has enough nonzero entries to ensure ergodicity (see below). The condition $p_g = p_{g^{-1}}$ is easily seen to be both necessary and sufficient to ensure detailed balance with respect to the equal-weight distribution π.

Consider, for example, the case $d = 2$. Then G is the dihedral group D_4, which has eight elements:

- identity (1)
- $\pm 90^\circ$ rotations (2)
- 180° rotation (1)
- axis reflections (2)
- diagonal reflections (2)

The condition $p_g = p_{g^{-1}}$ reduces to $p_{+90^\circ\text{rot}} = p_{-90^\circ\text{rot}}$. It will be proven in Section 3.5 that a sufficient condition for ergodicity is the nonvanishing of the probabilities p_g for

$$\left\{\begin{array}{c} \textit{either} \\ \pm 90^\circ \text{ rotations} \\ \textit{or} \\ \text{both diagonal reflections} \end{array}\right\} \quad \text{and} \quad \left\{\begin{array}{c} \textit{either} \\ 180^\circ \text{ rotation} \\ \textit{or} \\ \text{both axis reflections} \end{array}\right\}$$

In our simulations we have used the choice $p_{id} = 0$, $p_g = 1/7$ for $g \neq id$.

For $d = 3$, G is the octahedral group O_h, which has 48 elements. This group includes (among other things) rotations and reflections similar to those in $d = 2$, as well as 180° rotations about the face diagonal of a unit cube and $\pm 120^\circ$ rotations about the body diagonal of a unit cube.[31] In any case, a simple description of the symmetry group of $\mathbb{Z}^d$, valid for all d, goes as follows: An element $g \in G$ is a $d \times d$ orthogonal matrix with integer entries; so it suffices to specify the columns of g, which are $ge_1, ge_2, \ldots, ge_d$, where $e_1, e_2, \ldots, e_d$ are the unit vectors in $\mathbb{Z}^d$. Hence, an element $g \in G$ can be specified uniquely by giving a permutation π of $\{1, \ldots, d\}$ and numbers $\sigma_1, \ldots, \sigma_d = \pm 1$, and setting

$$ge_i = \sigma_i e_{\pi(i)} \tag{3.1}$$

It follows that the cardinality of G is $2^d d!$. Using this description of G, the pivot algorithm can be programmed very easily in any dimension.

Some variants on the pivot algorithm are worth mentioning. For example, the original algorithm of Lal[14] uses a *step* of the walk (rather than a *site*) as the pivot location, and reflects the part of the walk subsequent to that step in a hyperplane containing the chosen step. For the two-dimensional hexagonal lattice (the case considered by Lal), we do not know whether this algorithm is ergodic. However, for the simple (hyper)cubic lattice (any $d \geqslant 2$) or the triangular lattice, this algorithm is clearly *not* ergodic, since a straight rod is "frozen."

Another variant of the pivot algorithm on the square lattice is the following: Pick a pivot point k at random. If the angle of the SAW at ω_k is 180°, then attempt a diagonal reflection (choose one of the two diagonal reflections at random, with equal probability). On the other hand, if the angle at ω_k is 90°, then attempt either the diagonal reflection that would straighten this angle or else the axis reflection through the line determined by ω_k and ω_{k-1} (again with equal probability). It is easy to check that detailed balance holds. Ergodicity of this algorithm follows from the proof of Theorem 1 in Section 3.5. This method yields a higher acceptance fraction than the usual pivot algorithm, since it uses local information to choose its pivots.

For simplicity we have described the pivot algorithm as acting always on the part of the walk *subsequent* to the pivot point. However, the computational work can be reduced (roughly by a factor of 2) by applying the symmetry operation always to the *shorter* of the two segments of the walk, whichever it may be. In this variant of the algorithm, the initial point of the walk is no longer maintained fixed at the origin, so the foregoing description with state space $\mathscr{S}_N$ is inappropriate. Rather, the state space of the algorithm should be considered to be the space of *equivalence classes* of N-step SAWs modulo translation; the algorithm is then easily proven to be equivalent to the standard pivot algorithm. In practice, it is necessary to translate the walk back to the origin every once in a while in order to avoid integer overflow in the walk coordinates.

3.2. Acceptance Fraction and Autocorrelation Time

In this section we present a refined version of the heuristic argument sketched in the Introduction, which relates the autocorrelation time τ of the pivot algorithm to the acceptance fraction f. We then present a heuristic argument that attempts to predict the critical exponent p for the acceptance fraction ($f \sim N^{-p}$); and we compare this prediction to numerical estimates of p from exact enumeration (Appendix A) and Monte Carlo (Section 4.2). The predicted value of p turns out to be incorrect, but "in the right ballpark."

Suppose we know that the acceptance fraction f in the pivot algorithm behaves as $f \sim N^{-p}$ as $N \to \infty$. How, then, should we expect the autocorrelation time to behave? Note first that if the acceptance fraction is f, then, on the average, once every $1/f$ attempted moves we will obtain a success. Note also that the pivot moves are very radical: after a few (say, five or ten) successful pivots, the *global* conformation of the walk should have reached an "essentially new" state. Thus, we expect that for observables A that measure the *global* properties of the walk—such as the squared end-to-end distance ω_N^2 or the squared radius of gyration S_N^2—the autocorrelation time $\tau_{\text{int},A}$ should be a few (perhaps five) times $1/f$. On the other hand, it is important to recognize that *local* observables—such as the angle between the 17th and 18th steps of the walk—may evolve a factor of N more slowly than global observables. For example, the observable mentioned in the preceding sentence changes only when ω_{17} serves as a successful pivot point; and this happens, on average, only once every N/f attempted moves.[8] Thus, for *local* observables A we expect $\tau_{\text{int},A}$ to be of order N/f. By (2.23), τ_{exp} must be of at least this order; and if we have not overlooked any slow modes in the system, then τ_{exp} should be of exactly this order. Finally, even the global observables are unlikely to be precisely orthogonal [in $l^2(\pi)$] to the slowest mode; so it is reasonable to expect that $\tau_{\text{exp},A}$ be of order N/f for these observables, too. In other words, for global observables A we expect the autocorrelation function $\rho_{AA}(t)$ to have an extremely slowly decaying tail, which, however, contributes little to the area under the curve. This behavior is illustrated by the exact solution of the pivot dynamics for the case of ordinary random walk (Section 3.3).

The foregoing heuristic argument is, of course, far from a rigorous proof. It is not in general possible to find upper bounds on the autocorrelation time in terms of the acceptance fraction; the problem is that the state space could contain "bottlenecks" through which passage is unusually difficult. We have no reason to believe that such bottlenecks occur in the pivot algorithm, but neither do we have any proof of their nonexistence.

The heuristic argument is inaccurate in one additional respect. In Section 3.3 we will compute an *exact* solution for the pivot dynamics in the case of *ordinary* random walk (i.e., no self-avoidance constraint); the result is $f=1$, $\tau_{\text{exp}}=\tau_{\text{exp},A}\sim N$, but (for global observables A) $\tau_{\text{int},A}\sim\log N$. Thus, $\tau_{\text{int},A}$ is greater by a factor of $\log N$ than our naive argument would indicate. A similar factor of $\log N$ could conceivably occur in the self-avoiding case as well, so that we would have $f\sim N^{-p}$ but $\tau_{\text{int},A}\sim N^p\log N$. However, the numerical evidence presented in Sections 4.2 and 4.3 appears not to support this possibility.

Next we attempt to estimate heuristically the acceptance fraction f, and in particular to predict (at least approximately) the critical exponent p

[8] This important fact about the pivot algorithm was noticed by García de la Torre *et al.* (Ref. 32, p. 149, second column).

($f \sim N^{-p}$). Let $1 \leqslant k \leqslant N-1$ and let $g \in G$. Then the acceptance fraction for applications of symmetry operation g at pivot point k is some number $f(g, k, N)$: it is the fraction of walks $\omega \in \mathscr{S}_N$ for which all the transformed points $\omega_i' = \omega_k + g(\omega_i - \omega_k)$, $k+1 \leqslant i \leqslant N$, are disjoint from the points $\omega_0, \ldots, \omega_k$. The acceptance fraction f is then the average of $f(g, k, N)$ with respect to the probabilities $\{p_g\}_{g \in G}$ and $p_0, \ldots, p_{N-1}$.

A crude heuristic argument for the acceptance fraction $f(g, k, N)$ is the following: Suppose that the segments $\omega_0, \ldots, \omega_k$ and $\omega_k, \ldots, \omega_N$ of the walk ω behave as if they were typical k-step and $(N-k)$-step SAWs, respectively, in random relative orientation. In that case, the acceptance fraction would be precisely

$$f(g, k, N) = c_N / c_k c_{N-k} \tag{3.2}$$

since this is the probability that the joining of a random k-step SAW and a random $(N-k)$-step SAW results in an N-step SAW. Averaging over k (with respect to any reasonable distribution) and using the asymptotic behavior (2.2) of c_N, we predict

$$f \sim N^{-(\gamma-1)} \tag{3.3}$$

Hence, $p = \gamma - 1$. (Recall that $\gamma - 1$ is believed to equal $11/32 = 0.34375$ in $d=2$,[33] ≈ 0.162 in $d=3$,[34] and 0 in $d \geqslant 4$.)

Of course, the supposition on which this argument is based is wrong, for two reasons. First, the two segments of the walk ω are *not* typical k-step and $(N-k)$-step SAWs: the fact that they are known to occur on an N-step SAW means that they are somewhat "longer and thinner" than typical k-step and $(N-k)$-step SAWs, hence harder to intersect with. This property is preserved by the symmetry operation g, so one might expect that the acceptance fraction $f(g, k, N)$ would be *greater* than the prediction (3.2). On the other hand, the two segments are not in a random relative orientation: the fact that they are the two segments of an N-step SAW means that they are more likely to point in opposite directions (looking outward from the pivot point k), since this helps them to avoid intersecting each other. This nonrandom relative orientation certainly affects the acceptance fraction, since it is "remembered" by the transformed walk. Consider, for example, the trivial case in which the group element g is the identity element. Then the acceptance fraction is 1, much larger than predicted in (3.2), due to a combination of the "longer-and-thinner" effect and the "orientation" effect. (The acceptance fraction in this case is of course irrelevant for determining the autocorrelation time, since proposed "moves" with $g = id$ have the same effect whether they are "accepted" or "rejected": the system does not move at all.) On the other hand, if the group element g is a 180° rotation or a reflection (with certain axes), then one might expect the "orientation" effect to *reduce* the acceptance fraction below that predicted in (3.2): if the original orientation of the two segments tends to be antiparallel (so as to avoid intersection), then the new (rotated or reflected) orientation of the two segments may tend to be parallel (and thus favor intersection). For 180° rotations or reflections with other axes, or for 90° rotations, the situation is less clear.

In summary, the prediction (3.2) is a very crude estimate and is wrong for at least two reasons, leading to errors of possibly opposite signs. Hence, there is no reason whatsoever to believe that the critical exponent p is *exactly* equal to $\gamma - 1$ (except perhaps in dimension $d \geqslant 4$, where we expect $p = \gamma - 1 = 0$). On the other hand, this heuristic argument probably does capture the main qualitative features of the problem, so it *is* reasonable to expect that f does behave as $\sim N^{-p}$ with p equal to a small, positive number.

We have tested this heuristic argument in two ways:

1. We have performed an exact enumeration of SAWs in $d = 2$ up to $N = 17$, and computed exactly the acceptance fractions $f(g, k, N)$. These data are reported in Appendix A, where we also perform a "series-extrapolation" analysis. The results of this analysis are not overwhelmingly stable, but they yield the following rough estimates:

90° rotations:	$p = 0.145 \pm 0.04$
axis reflections:	$p = 0.175 \pm 0.04$
diagonal reflections:	$p = 0.165 \pm 0.04$
180° rotations:	$p \approx 0.41$ (not well converged)
group average:	$p = 0.18 \pm 0.04$

(95% subjective confidence limits). We find the radically different exponent for 180° rotations extremely surprising: it is natural to expect the acceptance fraction for 180° rotations to be much lower than that for other group elements, due to the "orientation" effect noted above; but, by standard ideas about universality, one would normally expect this to affect the amplitude and not the critical exponent. Initially we suspected, therefore, that the apparently larger exponent p for 180° rotations was a numerical artifact, which would disappear at larger values of N.

2. Our Monte Carlo runs yield extremely precise estimates of the acceptance fraction in $d = 2$ for a variety of values of N in the range $200 \leqslant N \leqslant 10000$. The peculiar results of the series analysis led us to reanalyze our Monte Carlo data so as to extract also (where possible) the acceptance

fractions broken down by symmetry-group elements. These data, reported in Section 4.2 (Tables II and III), show that $f \sim N^{-p}$ with

90° rotations:	$p = 0.1637 \pm 0.0020$
axis reflections:	$p = 0.1967 \pm 0.0021$
diagonal reflections:	$p = 0.1953 \pm 0.0021$
180° rotations:	$p = 0.505 \pm 0.03$ (somewhat subjective)
group average:	$p = 0.1926 \pm 0.0008$

(95% confidence limits). These data make it clear that the much larger critical exponent for 180° rotations is *not* a numerical artifact from small N, but is a real effect. It would thus be reasonable to expect that *all four* group elements (more precisely, conjugacy classes) might have distinct critical exponents p; and there is some support for this in the data for 90° rotations. If this is the case, then the exponent for the group average would be the *smallest* of the four individual exponents; but it would be afflicted by an unbelievably small correction-to-scaling exponent Δ_1 (≈ 0.03 according to our estimates), which would make accurate estimation of the exponent almost impossible (as our somewhat contradictory estimates show). In any case, we still do not understand *physically* why the acceptance-fraction exponent should be different for the group elements.

Our Monte Carlo data also provide somewhat less precise estimates for the autocorrelation times $\tau_{\text{int},A}$ of various global observables A (see Table IV and Section 4.3). Straight power-law fits yield $\tau_{\text{int},A} \sim N^q$ with $q \approx 0.205 \pm 0.015$. The data appear *not* to be consistent with the logarithmic behavior $\tau_{\text{int},A} \sim N^q \log N$ if we insist that $q \geqslant p$. Since the only theoretical reason for considering multiplicative logarithmic corrections was based on the idea that $q = p$, and since furthermore $q < p$ is highly implausible, we conclude that such logarithmic corrections are probably not present in the pivot algorithm for the $d = 2$ *self-avoiding* walk.

In summary, we find that in $d = 2$, $f \sim N^{-p}$ with $p \approx 0.19$. Clearly, p is *not* equal to $\gamma - 1 = 0.34375$, but is in fact somewhat smaller. However, we do confirm that $\tau_{\text{int},A} \sim N^q$ for global observables A, with $q \approx p$. Our preliminary results for $d = 3$ show the same qualitative behavior: $p \approx 0.107$ (for the group average) versus[34] $\gamma - 1 \approx 0.162$. Michael Fisher (private communication) has posed the following very interesting problem: *Express the critical exponent(s) p in terms of other critical exponents for the SAW* (many of which have recently been computed in $d = 2$[35,36]). This problem is completely open.

3.3. Exact Solution of Pivot Dynamics for Ordinary Random Walk

In this section we solve exactly for some of the dynamical properties of the pivot algorithm for the case of *ordinary* random walk (i.e., for walks without the self-avoidance constraint). First we compute the eigenvalues of the transition matrix P (and thus τ_{exp}); next we compute the autocorrelation functions $\rho_{AA}(t)$ (and thus $\tau_{\text{int},A}$) for selected observables A.

Consider, for starters, the case of ordinary random walk on a *two-dimensional* regular lattice, which we take to be either square (coordination number $q=4$), triangular ($q=6$), or hexagonal ($q=3$). Then we can represent an N-step walk ω by a sequence of integers $(l_1,..., l_N)$ with $1 \leqslant l_i \leqslant q$: the integer l_i codes the *angle* at ω_{i-1}, i.e., the signed angle between the $(i-1)$th and ith steps of the walk. (For $i=1$, l_i codes the *absolute* orientation of the first step of the walk.) The configuration space of N-step walks is thus a product space $S=\{1,..., q\}^N$.

Let us consider, to begin with, pivot algorithms that use only *rotations*, not reflections. We also assume (for simplicity only) that the pivot point k is chosen from a *uniform* distribution over $0 \leqslant k \leqslant N-1$. Then the transition matrix P is of the form

$$P=\frac{1}{N}\sum_{i=1}^{N} I^{\otimes i-1} \otimes R \otimes I^{\otimes N-i} \tag{3.4}$$

where I is the $q \times q$ identity matrix, and R is a fixed $q \times q$ symmetric stochastic matrix (the details of R depend on the choice of the probabilities $\{p_g\}_{g \in G}$; the symmetry is a consequence of detailed balance). The eigenvalues of R are *real*; we denote them $1 \equiv \mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_q \geqslant -1$. The eigenvalues of P are thus all of the form

$$\lambda=\frac{1}{N}\sum_{\alpha=1}^{q} n_\alpha \mu_\alpha \tag{3.5a}$$

where $n_1,..., n_q$ are nonnegative integers satisfying $\sum_{\alpha=1}^{q} n_\alpha = N$ (they are the "occupation numbers" of the various eigenstates); the multiplicity of this eigenvalue is

$$N! \Big/ \prod_{\alpha=1}^{q} n_\alpha! \tag{3.5b}$$

So the eigenvalues of P are rather uniformly distributed (with, however, nonconstant multiplicity) between $\lambda_1 = \mu_1 = 1$ and $\lambda_{q^N} = \mu_q$. In particular, the next-to-leading eigenvalue of P is $\lambda_2 = 1-(1-\mu_2)/N = 1-O(N^{-1})$.

This means that the slowest mode in the system[9] has relaxation time $\tau_{\exp} = -(\log \lambda_2)^{-1} \sim N$.

For example, if the probabilities $\{p_g\}$ are chosen to be uniform over the rotation subgroup of G, then we have $R = (1/q)E$, where E is the $q \times q$ matrix with all entries equal to 1. The eigenvalues of R are $\mu_1 = 1$, $\mu_2 = \cdots = \mu_q = 0$, and the next-to-leading eigenvalue of P is $\lambda_2 = 1 - 1/N$. To take another example, suppose that the probabilities $\{p_g\}$ are chosen to be uniform over the *nonidentity* elements of the rotation subgroup of G. Then we have

$$R = \frac{1}{q-1} E - \frac{1}{q-1} I$$

The eigenvalues of R are $\mu_1 = 1$, $\mu_2 = \cdots = \mu_q = -1/(q-1)$, and the next-to-leading eigenvalue of P is

$$\lambda_2 = 1 - \left(\frac{q}{q-1}\right) \frac{1}{N}$$

If *reflections* as well as rotations are allowed, then the transition matrix becomes more complicated. A reflection acting at the ith coordinate (i.e., at pivot point ω_{i-1}) changes not only l_i, but also $l_{i+1}, \ldots, l_N$, since these latter angles are *reversed in sign* ($\theta \to -\theta$). Call this sign-reversal operation J; note that J is a permutation matrix. Then the transition matrix for a reflection acting at the ith coordinate is of the form $I^{\otimes i-1} \otimes R' \otimes J^{\otimes N-i}$, where R' is a suitable $q \times q$ matrix (which depends on the relative probabilities assigned to the different reflections). Hence the full transition matrix P is of the form

$$P = \frac{\alpha}{N} \sum_{i=1}^{N} I^{\otimes i-1} \otimes R \otimes I^{\otimes N-i} + \frac{1-\alpha}{N} \sum_{i=1}^{N} I^{\otimes i-1} \otimes R' \otimes J^{\otimes N-i} \qquad (3.6)$$

where α is a constant ($0 \leqslant \alpha \leqslant 1$) which expresses the relative probability of rotations and reflections. We do not know how to diagonalize this P in the most general case. But if both R and R' are linear combinations of E and I (as in the two examples above), then all of the matrices involved here can be simultaneously diagonalized, since J commutes with E. In a suitable basis, we have $E = \operatorname{diag}(q, 0, \ldots, 0)$ and $J = \operatorname{diag}(1, \ldots, 1, -1, \ldots, -1)$. Then, for any given R and R', the eigenvalues of P can be computed as before. Clearly we will again have $\lambda_2 = 1 - O(1/N)$.

[9] We assume here that the algorithm is aperiodic, i.e., that $\mu_q > -1$. This ensures that, for large N, $\lambda_2 > |\lambda_{q^N}| = |\mu_q|$.

We do not know how to extend the above exact analysis to dimensions $d \geqslant 3$. However, in Appendix B we prove upper and lower bounds showing that the result $\lambda_2 = 1 - O(1/N)$ continues to hold.

Now we turn to the calculation of the autocorrelation function $\rho_{AA}(t)$ for certain global observables A. We consider the pivot algorithm on $\mathbb{Z}^d$ for arbitrary dimension d. For simplicity we consider only the case in which the probabilities $\{p_g\}$ are uniformly distributed over the group G (*including* the identity element), i.e., $R = R' = (1/q)E$ and $\alpha = 1/2$ (but not all these conditions are really necessary for our analysis to hold). We define $\mathbf{a}_i$ to be the vector corresponding to the ith step of the walk, i.e., $\mathbf{a}_i = \omega_i - \omega_{i-1}$. Then the cross-correlation function of $\mathbf{a}_i$ and $\mathbf{a}_j$ is

$$\begin{aligned}\langle \mathbf{a}_i(0) \cdot \mathbf{a}_j(t) \rangle &= \delta_{ij} \times \text{Prob(in } t \text{ trials no pivot point } < i \text{ is chosen)} \\ &= \delta_{ij} \times (1 - i/N)^{|t|} \qquad (3.7)\end{aligned}$$

(Note that $\mathbf{a}_1$ couples to the slowest mode $\lambda_2 = 1 - 1/N$.) It follows that the (unnormalized) autocorrelation function of $\omega_N \equiv \sum_{i=1}^{N} \mathbf{a}_i$ is

$$C_{\omega_N, \omega_N}(t) \equiv \langle \omega_N(0) \cdot \omega_N(t) \rangle = \sum_{i=1}^{N} \left(1 - \frac{i}{N}\right)^{|t|} \qquad (3.8)$$

This is the spectral representation, guaranteed on general grounds by (2.21); note that it has a uniform spectrum of contributions, ranging from the slowest exponential ($i = 1$) to the fastest ($i = N$). Thus,

$$\tau_{\text{exp},\omega_N} = -1/\log(1 - 1/N) \sim N \qquad \text{as} \quad N \to \infty$$

On the other hand, the *integrated* autocorrelation time for ω_N is

$$\begin{aligned}\tau_{\text{int},\omega_N} \equiv \frac{1}{2} \sum_{t=-\infty}^{\infty} \rho_{\omega_N, \omega_N}(t) &= \sum_{i=1}^{N} \left(\frac{1}{i} - \frac{1}{2N}\right) \\ &= \log N + \left(C - \frac{1}{2}\right) + O(N^{-1}) \qquad (3.9)\end{aligned}$$

where C is Euler's constant. Thus, $\tau_{\text{int},\omega_N} \sim \log N$ as $N \to \infty$. Finally, we note that the autocorrelation function $\rho_{\omega_N, \omega_N}(t)$ has *two distinct scaling behaviors* as $N \to \infty$, depending on the regime of t:

$$\rho_{\omega_N, \omega_N}(t) \approx \begin{cases} \dfrac{1}{|t| + 1} & \text{if} \quad |t| \ll N \\[2ex] \dfrac{1}{N} \dfrac{e^{-|t|/N}}{1 - e^{-|t|/N}} & \text{if} \quad |t| \sim N \end{cases} \qquad (3.10)$$

(More precisely, the first expression is valid for $N \to \infty$ with $t/N \to 0$; the second expression is valid for $N \to \infty$ with $|t|/N \geqslant \varepsilon > 0$.)

Next we look at the observable $\mathbf{a}_i \cdot \mathbf{a}_j$ $(i < j)$, which is the cosine of the angle between the ith and jth steps of the walk. The mean value $\langle \mathbf{a}_i \cdot \mathbf{a}_j \rangle$ is zero (by symmetry). The cross-correlation function of $\mathbf{a}_i \cdot \mathbf{a}_j$ with $\mathbf{a}_k \cdot \mathbf{a}_l$ is

$$\begin{aligned} &\langle \mathbf{a}_i \cdot \mathbf{a}_j(0)\, \mathbf{a}_k \cdot \mathbf{a}_l(t) \rangle \\ &\quad = \frac{1}{d} \delta_{ik} \delta_{jl} \times \text{Prob(in } t \text{ trials no pivot point in the interval } [i, j) \text{ is chosen)} \\ &\quad = \frac{1}{d} \delta_{ik} \delta_{jl} \times \left(1 - \frac{j-i}{N}\right)^{|t|} \qquad (3.11) \end{aligned}$$

[Reason: Note first that the $\{\mathbf{a}_i(0)\}_{1 \leqslant i \leqslant N}$ are independent random vectors taking the values $\pm e_1, \pm e_2, \ldots, \pm e_d$ with probability $1/2d$; it easily follows that

$$\langle \mathbf{a}_i \cdot \mathbf{a}_j(0)\, \mathbf{a}_k \cdot \mathbf{a}_l(0) \rangle = \frac{1}{d} \delta_{ik} \delta_{jl}$$

for $i < j$, $k < l$. Now consider the expectation of $\mathbf{a}_k \cdot \mathbf{a}_l(t)$ conditioned on the configuration at time 0 and also conditioned on the occurrence or not of a pivot site in the interval $[k, l)$ sometime during the time interval $(0, t]$. If no such pivot occurred, then $\mathbf{a}_k \cdot \mathbf{a}_l(t) = \mathbf{a}_k \cdot \mathbf{a}_l(0)$. If such a pivot occurred, then $\mathbf{a}_k(t)$ and $\mathbf{a}_l(t)$ have random relative orientations, so the conditional mean of $\mathbf{a}_k \cdot \mathbf{a}_l(t)$ is zero.] It follows that the (unnormalized) autocorrelation function of

$$\omega_N^2 \equiv \sum_{i,j=1}^{N} \mathbf{a}_i \cdot \mathbf{a}_j = N + 2 \sum_{1 \leqslant i < j \leqslant N} \mathbf{a}_i \cdot \mathbf{a}_j$$

is

$$\begin{aligned} C_{\omega_N^2, \omega_N^2}(t) \equiv \langle \omega_N^2(0) \cdot \omega_N^2(t) \rangle - \langle \omega_N^2 \rangle^2 &= \frac{4}{d} \sum_{1 \leqslant i < j \leqslant N} \left(1 - \frac{j-i}{N}\right)^{|t|} \\ &= \frac{4N}{d} \sum_{m=1}^{N-1} \left(1 - \frac{m}{N}\right)^{|t|+1} \qquad (3.12) \end{aligned}$$

The slowest decaying contribution comes from $m = 1$; hence

$$\tau_{\exp, \omega_N^2} = -1/\log(1 - 1/N) \sim N$$

as $N \to \infty$. Moreover,

$$C_{\omega_N^2, \omega_N^2}(0) \equiv 2N(N-1)/d$$

so the normalized autocorrelation function is

$$\rho_{\omega_N^2, \omega_N^2}(t) = \frac{2}{N-1} \sum_{m=1}^{N-1} \left(1 - \frac{m}{N}\right)^{|t|+1} \tag{3.13}$$

Hence, the *integrated* autocorrelation time for ω_N^2 is

$$\begin{aligned} \tau_{\mathrm{int}, \omega_N^2} &\equiv \frac{1}{2} \sum_{t=-\infty}^{\infty} \rho_{\omega_N^2, \omega_N^2}(t) \\ &= \frac{1}{N-1} \sum_{m=1}^{N-1} \left(1 - \frac{m}{N}\right)\left(\frac{2N}{m} - 1\right) \\ &= \frac{1}{N-1} \left[2N \log N + \left(2C - \frac{5}{2}\right) N + O(1)\right] \\ &= 2 \log N + \left(2C - \frac{5}{2}\right) + O\left(\frac{\log N}{N}\right) \end{aligned} \tag{3.14}$$

Thus,

$$\tau_{\mathrm{int}, \omega_N^2} \sim \log N \qquad \text{as} \quad N \to \infty$$

Finally, we note that the autocorrelation function $\rho_{\omega_N^2, \omega_N^2}(t)$ has two distinct scaling behaviors as $N \to \infty$, depending on the regime of t:

$$\rho_{\omega_N^2, \omega_N^2}(t) \approx \begin{cases} \dfrac{2}{|t| + 2} & \text{if} \quad |t| \ll N \\ \dfrac{2}{N} \dfrac{e^{-|t|/N}}{1 - e^{-|t|/N}} & \text{if} \quad |t| \sim N \end{cases} \tag{3.15}$$

We can also use (3.11) to compute the autocorrelation function of S_N^2, the squared radius of gyration. From (2.6) we find

$$S_N^2 = \frac{N(N+2)}{6(N+1)} + \frac{2}{(N+1)^2} \sum_{1 \leqslant i < j \leqslant N} i(N+1-j)\, \mathbf{a}_i \cdot \mathbf{a}_j \tag{3.16}$$

Inserting this into (3.11), we obtain

$$\begin{aligned} C_{S_N^2, S_N^2}(t) &= \frac{4}{d(N+1)^4} \sum_{1 \leqslant i < j \leqslant N} i^2 (N+1-j)^2 \left(1 - \frac{j-i}{N}\right)^{|t|} \\ &= \frac{2}{15 d(N+1)^4} \sum_{l=1}^{N-1} l(l+1)(l+2)(l^2+2l+2) \left(\frac{l}{N}\right)^{|t|} \end{aligned} \tag{3.17}$$

A tedious calculation yields

$$C_{S_N^2,S_N^2}(0)=\frac{4}{d(N+1)^4}\frac{(N-1)\,N(N+1)(N+2)(2N^2+2N+3)}{360} \tag{3.18}$$

and an even more tedious calculation yields

$$\begin{aligned}\tau_{\mathrm{int},S_N^2} &\equiv \frac{1}{2}\sum_{t=-\infty}^{\infty} \rho_{S_N^2,S_N^2}(t)\\ &=\frac{6}{(N-1)\,N(N+1)(N+2)(2N^2+2N+3)}\\ &\quad\times\sum_{l=1}^{N-1} l(l+1)(l+2)(l^2+2l+2)\left(\frac{2N}{N-l}-1\right)\\ &=6\log N+\left(6C-\frac{71}{5}\right)+O\left(\frac{\log N}{N}\right)\end{aligned} \tag{3.19}$$

Thus,

$$\tau_{\mathrm{int},S_N^2} \sim \log N \qquad \text{as} \quad N \to \infty$$

Finally, we note that the autocorrelation function $\rho_{S_N^2,S_N^2}(t)$ has two distinct scaling behaviors as $N \to \infty$, depending on the regime of t:

$$\rho_{S_N^2,S_N^2}(t) \approx \begin{cases} \dfrac{6}{|t|+6} & \text{if} \quad |t| \ll N \\ \dfrac{6}{N}\dfrac{e^{-|t|/N}}{1-e^{-|t|/N}} & \text{if} \quad |t| \sim N \end{cases} \tag{3.20}$$

In summary, all three of the *global* observables $A=\omega_N$, ω_N^2, S_N^2 have $\tau_{\mathrm{exp},A} \sim N$, but $\tau_{\mathrm{int},A} \sim \log N$. On the other hand, *local* observables like $A=\mathbf{a}_i$ or $\mathbf{a}_i \cdot \mathbf{a}_j$ (*i, j fixed*) have $\tau_{\mathrm{exp},A} \sim \tau_{\mathrm{int},A} \sim N$.

Finally, the foregoing formulas also shed some light on the relative efficiency of $\langle \omega_N^2 \rangle$ and $\langle S_N^2 \rangle$ in making Monte Carlo estimates of ν. On the one hand, the relative variance of S_N^2 is (for large N) only 2/5 that of ω_N^2:

$$\frac{\mathrm{var}(S_N^2)}{\langle S_N^2 \rangle^2}=\frac{N^2/45d+O(N)}{N^2/36+O(N)}=\frac{4}{5d}+O\left(\frac{1}{N}\right) \tag{3.21}$$

$$\frac{\mathrm{var}(\omega_N^2)}{\langle \omega_N^2 \rangle^2}=\frac{2N(N-1)/d}{N^2}=\frac{2}{d}+O\left(\frac{1}{N}\right) \tag{3.22}$$

This is because the radius of gyration is a somewhat more "global" measure of the size of the walk than is the end-to-end distance. On the other hand, τ_{int,S_N^2} for the pivot algorithm is asymptotically three times as large as $\tau_{\text{int},\omega_N^2}$:

$$\frac{\tau_{\text{int},S_N^2}}{\tau_{\text{int},\omega_N^2}}=\frac{6\log N+(6C-71/5)+O((\log N)/N)}{2\log N+(2C-5/2)+O((\log N)/N)}$$
$$=3-\frac{67}{20(\log N+C-5/4)}+O\left(\frac{1}{N}\right)$$

(Note, however, that even for $N=10{,}000$ this ratio is only ≈ 2.61.) It follows from (2.19b) that the relative variance in estimates of $\langle S_N^2\rangle$ by the pivot algorithm will be asymptotically 6/5 as large as the relative variance in estimates of $\langle\omega_N^2\rangle$. For the self-avoiding walk, these constants will of course be changed, but we expect them to be "in the same ballpark"; and this is indeed the case (see Tables II and IV in Section 4.2).

3.4. Data Structures and Computational Complexity

In this section we discuss the data structures needed in implementing the pivot algorithm, and analyze the algorithm's computational complexity. We also make some practical remarks regarding implementation of the computer program.

We store the coordinates of the current walk $\omega=(\omega_0,\dots,\omega_N)$ using two (redundant) data structures: a sequentially allocated linear list and a hash table. The former is self-explanatory. A *hash table* can be defined abstractly as a data structure with the following properties: Given a finite (but typically enormous) set K of "possible keywords," we wish to store a subset $H\subset K$ (of cardinality $\leqslant$ some maximum M) in such a way that, for any $x\in K$, the following operations can be carried out rapidly:

1. *Query.* Is $x\in H$?
2. *Insertion.* Insert x into H (if it is not in H already).
3. *Deletion.* Delete x from H (if it is in H currently).

Specific implementations of the hash table will be discussed below, along with the precise meaning of the word "rapidly." Roughly speaking, "rapidly" means "in a time of order 1, on the average." In our application, the set K of possible keywords will be the set of all points in some box $B\subset\mathbb{Z}^d$ that is large enough to contain all possible points in the walk ω (e.g., a cube of side $\geqslant 2N$ centered at the origin).

Our sequentially allocated linear list is a permanent data structure, which contains at all times the current configuration of the walk ω. Our hash table is, on the other hand, a scratch data structure, which is used solely for self-avoidance checking; it is initialized to empty ($H=\varnothing$) and is reset to this condition after each use. We now describe in more detail how the self-avoidance checking is performed.

Suppose that a pivot site ω_k ($1 \leqslant k \leqslant N-1$) and a symmetry transformation g have been (randomly) chosen. We then have to compute the proposed new walk ω', defined by

$$\omega_i' = \begin{cases} \omega_i & \text{for} \quad 0 \leqslant i \leqslant k \\ \omega_k + g(\omega_i - \omega_k) & \text{for} \quad k+1 \leqslant i \leqslant N \end{cases} \tag{3.24}$$

and test whether $\omega_i' \neq \omega_j'$ for all $0 \leqslant i < k < j \leqslant N$. (If i, j are both $\leqslant k$ or both $\geqslant k$, then $\omega_i' \neq \omega_j'$ is guaranteed, since the original walk ω was self-avoiding.) Note now that *if there is an intersection, it is most likely to occur for i and j both close to k* (see below for details). So we check those positions first. That is, we enter the points ω_{i+1}', ω_{i-1}', ω_{i+2}', ω_{i-2}',..., in that order, into the scratch hash table, checking for repetitions. If a self-intersection is encountered, then the procedure is immediately terminated and the proposed pivot move is unsuccessful. If, on the other hand, the entire walk ω' is entered into the hash table without encountering an intersection, then the proposed pivot move is successful. In either case, the hash table is cleaned up before exit.

The computational complexity of this self-avoidance-checking algorithm can be analyzed as follows: Define $\omega'[i, j]$ to be the set $\{\omega'_{\max(i,0)}, \omega'_{\max(i,0)+1}, \ldots, \omega'_{\min(j,N)}\}$. Then one application of the self-avoidance-checking algorithm (including reinitialization) requires a time of order $I(\omega')$, where

$$I(\omega') \equiv \begin{cases} \min\{i\colon \omega'[k-i, k+i] \text{ is not a SAW}\} \\ \qquad \text{if} \quad \omega' \text{ is not a SAW} \\ N \qquad \text{if} \quad \omega' \text{ is a SAW} \end{cases} \tag{3.25}$$

In particular, the amount of time required is at most $O(N)$. Now, if the acceptance fraction is $\sim N^{-p}$, then we can expect a successful pivot once every $\sim N^p$ attempts; so the amount of work required per successful pivot is certainly at most $O(N^{1+p})$. But in fact we can improve this bound to $O(N)$, by the following heuristic argument:

As remarked above, the amount of work per attempt is of order $I(\omega')$. Let us estimate crudely the expected value $E[I(\omega')]$:

$$\begin{aligned} \operatorname{Prob}\{I(\omega') > i\} &= \operatorname{Prob}\{\omega'[k-i, k+i] \text{ is a SAW}\} \\ &\approx \operatorname{Prob}\{\text{a } 2i\text{-step SAW pivoted at } i \text{ is again a SAW}\} \\ &\sim i^{-p} \end{aligned} \tag{3.26}$$

Therefore,

$$\begin{aligned} E[I(\omega')] &= \sum_{j=0}^{N} j \operatorname{Prob}\{I=j\} \\ &= \sum_{i=0}^{N} \operatorname{Prob}\{I>i\} \\ &\sim N^{1-p} \end{aligned} \tag{3.27}$$

since $p<1$ (this is crucial). In particular, the conditional expected values given that the attempt succeeds or fails are

$$E[I \mid \text{success}] = N \tag{3.28}$$

$$E[I \mid \text{failure}] \sim N^{1-p} \tag{3.29}$$

Thus, the expected amount of work required per successful pivot is of order

$$\begin{aligned} & E[I \mid \text{failure}] \cdot E[\text{number of failures until a success occurs}] \\ & \quad + E[I \mid \text{success}] \\ & = O(N^{1-p}) \frac{1}{N^{-p}} + N \\ & = O(N) \end{aligned} \tag{3.30}$$

Strong numerical evidence confirming this prediction will be presented in Section 4.4 (see Table V).

Combining the above with the conjecture $\tau_{\text{int}} \sim N^p$ from Section 3.2 (omitting possible logarithms), it follows that the pivot algorithm requires $O(N)$ work per "effectively independent" observation of a global observable. To see how good this is, consider any other Monte Carlo algorithm (dynamic or static) for generating SAWs. To get an "effectively independent" data point for a global observable, it is necessary to change at least εN sites of the walk, for some fixed $\varepsilon > 0$. If each of these new sites is computed separately, then this algorithm would require at least order N work per "effectively independent" observation. Therefore, any algorithm that would surpass the order-N bound cannot afford to compute (or even store) most of the new sites on successive walks.[10]

The easiest way to implement the "abstract hash table" is as a "bit

[10] One might imagine storing a walk as a sequence of bond rotations rather than as a sequence of points (see Section 3.3 and Appendix B). In this formulation, a pivot move changes only one element. But it is hard to see how to check self-avoidance after a proposed pivot move without computing explicitly the walk coordinates.

map," i.e., a large array in which each keyword $x \in K$ (i.e., each point in the box B) is assigned one bit: this bit is set to 1 if $x \in H$, and 0 otherwise. Then the operations of query, insertion, and deletion can obviously be performed in a time of order 1. The chief drawback of this method is its extravagant space requirements: the array requires at least $(2N+1)^d$ bits. On most machines this is unfeasible even in $d=2$ if N exceeds a few thousand, or in $d \geqslant 3$ if N exceeds a few hundred.

An alternate implementation uses the method of *hash-coding*(37,38): An array of M words is assigned, and each keyword $x \in K$ is assigned a *primary address* $h(x)$ in this array. Since in general $M \ll |K|$, the "hash function" h is necessarily many-to-one, i.e., many distinct keywords may share the same primary address, leading to the possibility of *collisions*. The various hash-coding algorithms are distinguished by the method they use to resolve collisions, i.e., to decide where to store a keyword if its primary address happens to be occupied by some other keyword. One of the simplest collision-resolution schemes, and the one we use, is *linear probing*(37,38): if the primary address $h(x)$ is occupied, the algorithm searches successively in addresses $h(x)+1$, $h(x)+2$,... (modulo M) until it finds either the keyword x or an empty slot.

In the worst possible case, a single query or insertion into a hash table containing N entries could take a time of order N. However, it can be shown(37) that as long as the hash table does not get close to full (i.e., N does not get near M), then the *average* time (i.e., if the points are randomly distributed) for a single query or insertion is of order 1. So the hash-coding method is nearly as fast as the bit-map method, and far more space-effective.

We remark that *deletion* from a linear-probing hash table is a delicate affair: if done naively, entries can get "lost."(37) Fortunately, in our application deletions occur only when cleaning up the table at the end; therefore, all difficulties can be avoided either by performing these deletions in a last-in-first-out manner, or by keeping a list of the locations in which elements have been inserted and then deleting precisely these entries. (We did the latter.)

In choosing the hash function h, we want the image set $h[\omega']$ to be "sparse": that is, if ω'_i happens to be close to ω'_j, then we want $h(\omega'_i)$ to be far from $h(\omega'_j)$. This is particularly important, since the occupied lattice sites in a self-avoiding walk *are* close together. We used hash functions of the form

$$h(x_1,..., x_d) = (a_1 x_1 + \cdots + a_d x_d) \bmod M \tag{3.31}$$

where $a_1,..., a_d$, M are chosen to be relatively prime and satisfy

$$a_k \approx M^{k/(d+1)} \tag{3.32}$$

Thus, the a_k are all of different magnitude, which helps ensure the desired behavior of h. [To understand this, think about why $h(x_1,..., x_d) = (x_1 + \cdots + x_d) \bmod M$ is a bad hash function.] In particular, because we are using linear probing, we want to avoid near-collisions as well as collisions; this is why we insist on $a_k > 1$ for all k.

To test how well h "hashed" its input, we repeated some runs with different values of M between $2N$ and $10N$, each time choosing $a_1,..., a_d$ as specified above. The difference in overall running time was negligible—less than 3%. We concluded that relatively little time was being wasted due to $h(\omega_i')$ coinciding with $h(\omega_j')$ for $\omega_i' \neq \omega_j'$.

3.5. Proof of Ergodicity

In this section we prove the ergodicity of various versions of the pivot algorithm. We have tried hard to convey the main ideas of the proofs through pictures and informal descriptions preceding the formal arguments.

Theorem 1. The pivot algorithm is ergodic for self-avoiding walks on $\mathbb{Z}^d$ provided that all axis reflections, and either all 90° rotations or all diagonal reflections, are given nonzero probability. In fact, any N-step SAW can be transformed into a straight rod by some sequence of $2N-1$ or fewer such pivots.

Notation. Before we explain the ideas behind the proof, we need to establish some notation. We consider SAWs $\omega = (\omega_0, \omega_1,..., \omega_N)$ in $\mathbb{Z}^d$, where ω_0 is not necessarily 0. Let $X_j(\omega_k)$ denote the jth coordinate of ω_k, so that $\omega_k = (X_1(\omega_k), X_2(\omega_k),..., X_d(\omega_k))$. We define $B(\omega)$ to be the smallest rectangular box containing ω, that is,

$$B(\omega) = \{(x_1,..., x_d): m_j^1(\omega) \leqslant x_j \leqslant m_j^2(\omega) \text{ for all } j = 1,..., d\} \tag{3.33}$$

where

$$m_j^1(\omega) = \min\{X_j(\omega_k): k = 0, 1,..., N\} \tag{3.34}$$

and

$$m_j^2(\omega) = \max\{X_j(\omega_k): k = 0, 1,..., N\} \tag{3.35}$$

are the minimum and maximum values, respectively, of the jth coordinate. A "face" of $B(\omega)$ is any set of the form $\{x \in B(\omega): x_j = m_j^i(\omega)\}$ for some $i = 1, 2$ and some $j = 1,..., d$. Let

$$M_j(\omega) = m_j^2(\omega) - m_j^1(\omega) \tag{3.36}$$

and let

$$D(\omega) = M_1(\omega) + \cdots + M_d(\omega) \tag{3.37}$$

Thus, $M_j(\omega)$ is the extension of the walk ω in the jth coordinate direction, and $D(\omega)$ is the l^1 diameter of $B(\omega)$. Finally, let $A(\omega)$ be the number of straight internal angles of ω:

$$A(\omega) = \#\{k: 0 < k < N \text{ and } \omega_k = \tfrac{1}{2}(\omega_{k-1} + \omega_{k+1})\} \tag{3.38}$$

Proof of Theorem 1. Here is the plan of the proof: We will partition the set of all N-step SAWs into two subsets:

(a) The set of those ω for which some face of $B(\omega)$ contains neither of the endpoints ω_0, ω_N.

(b) The set of those ω for which the endpoints ω_0 and ω_N are in opposite corners of $B(\omega)$.

(Every N-step SAW lies in exactly one of these two subsets.) If ω is in subset (a), we will show that there exists a pivot point ω_t and an axis reflection whose result is a SAW ω' with $D(\omega') > D(\omega)$ and $A(\omega') = A(\omega)$. If ω is in subset (b) and is not a straight rod, we will show that there exists a pivot point ω_s and a 90° rotation (or a diagonal reflection) whose result is a SAW ω' with $A(\omega') = A(\omega) + 1$ and $D(\omega') \geqslant D(\omega)$. From this, we conclude that for every N-step SAW ω that is not a rod, there exists a SAW ω' that may be obtained from ω by a single pivot, and satisfies $A(\omega') + D(\omega') > A(\omega) + D(\omega)$. Since $0 \leqslant A \leqslant N-1$ and $0 \leqslant D \leqslant N$ for every N-step SAW, and $A + D = 2N - 1$ if and only if the walk is a rod, it follows that any N-step SAW can be transformed into a rod by a sequence of at most $2N - 1$ pivots.

Now for the proof. First suppose that ω is a SAW in subset (a), i.e., suppose that there exists a coordinate hyperplane $x_j = c$ which determines a face of $B(\omega)$, such that neither ω_0 nor ω_N lies in this hyperplane. Then we will show that we can perform a successful pivot that is a reflection through this hyperplane; the pivot point ω_t is chosen to be the *first* point of ω that lies in this hyperplane. (See Fig. 1.) The resulting SAW, ω', will be seen to satisfy $A(\omega') = A(\omega)$, $M_j(\omega') > M_j(\omega)$, and $M_l(\omega') = M_l(\omega)$ for $l \neq j$, hence $D(\omega') > D(\omega)$.

In detail: Suppose that there exist $i \in \{1, 2\}$ and $j \in \{1,..., d\}$ such that neither ω_0 nor ω_N lies in the face $\{x \in B: x_j = m_j^i\}$. Let $t = \min\{k: X_j(\omega_k) = m_j^i\}$. Now reflect $\omega_{t+1},..., \omega_N$ through the hyperplane $x_j = m_j^i$, yielding the walk $\omega' = (\omega'_0,..., \omega'_N)$ defined by

$$\text{for} \quad k \leqslant t, \qquad \omega'_k = \omega_k \tag{3.39a}$$

$$\text{for} \quad k > t, \qquad X_l(\omega'_k) = X_l(\omega_k) \quad \text{for } l \neq j \tag{3.39b}$$

$$X_j(\omega'_k) = 2m_j^i - X_j(\omega_k) \tag{3.39c}$$

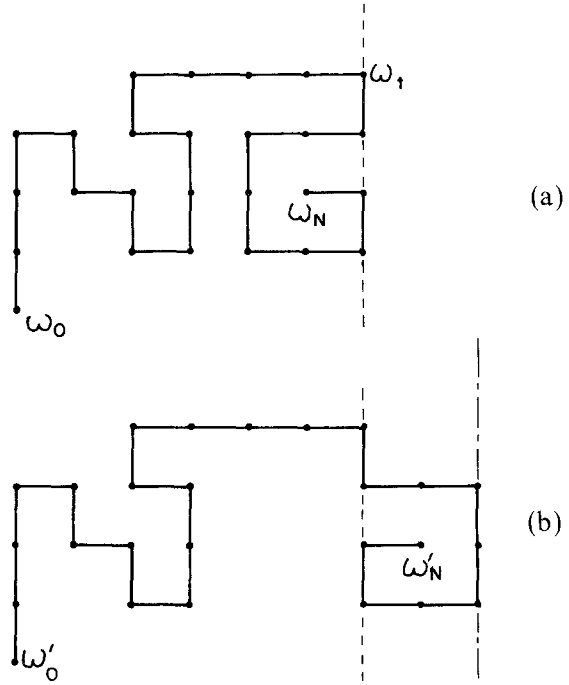

Fig. 1. (a) Pivot at ω_t by reflecting through the dashed line. (b) The result. (We can now reflect through the dashed-and-dotted line.)

It is not hard to see that ω' is indeed a SAW. We have to check that for all $k > t$, ω_k is not in the set $\{\omega_0,..., \omega_t\}$. This is clear if $X_j(\omega_k) \neq m_j^i$, since then $\omega'_k \notin B(\omega)$; and if $X_j(\omega_k) = m_j^i$, then $\omega'_k = \omega_k$, so the result follows because ω was a SAW.

Also, $A(\omega') = A(\omega)$, because right angles are preserved by axis reflections. [Note that $X_j(\omega_{t-1}) \neq m_j^i$, but $X_j(\omega_{t+1}) = m_j^i$, so both ω and ω' have right angles at ω_t.]

Next, we show that $D(\omega') > D(\omega)$. First, it is clear that $M_l(\omega') = M_l(\omega)$ for $l \neq j$. Now, let $Q_{r,s}(\omega)$ be the extension in the jth coordinate direction of the subwalk $(\omega_r, \omega_{r+1},..., \omega_s)$, i.e.,

$$Q_{r,s}(\omega) = \max\{X_j(\omega_k): r \leqslant k \leqslant s\} - \min\{X_j(\omega_k): r \leqslant k \leqslant s\} \tag{3.40}$$

Then

$$M_j(\omega) = \max(Q_{0,t}(\omega), Q_{t,N}(\omega)) \tag{3.41}$$

while

$$M_j(\omega') = Q_{0,t}(\omega) + Q_{t,N}(\omega) \tag{3.42}$$

Both $Q_{0,t}(\omega)$ and $Q_{t,N}(\omega)$ are strictly positive [since $X_j(\omega_0) \neq m_j^i$ and $X_j(\omega_N) \neq m_j^i$], so $M_j(\omega') > M_j(\omega)$. This completes the proof in case (a).

Next, suppose that ω is a SAW that is in subset (b) and is not a rod. Then $A(\omega) < N-1$ (i.e., ω contains at least one right angle), so we choose our pivot point ω_s to be the last right angle of ω. Since ω_N lies in a corner of $B(\omega)$, there must be a face of ω that contains $\omega_s, \omega_{s+1},..., \omega_N$ but does *not* contain ω_{s-1}. (See Fig. 2a.) We now perform a 90° rotation (or diagonal reflection) with pivot point ω_s so as to straighten out the angle at ω_s. In the resulting walk ω', points $\omega_{s+1}, \omega_{s+2},..., \omega_N$ will lie outside $B(\omega)$. The result will be an increase by 1 of $A(\omega)$; D cannot decrease, since one M_j will increase by $N-s$, another will decrease by at most $N-s$, and the rest will not change at all.

Formally, ω is in subset (b) if for each $j \in \{1,..., d\}$, either $X_j(\omega_0) = m_j^1(\omega)$ and $X_j(\omega_N) = m_j^2(\omega)$, or else $X_j(\omega_0) = m_j^2(\omega)$ and $X_j(\omega_N) = m_j^1(\omega)$. Let

$$s = \max\{k\colon 0 < k < N \text{ and } \omega_k \neq \tfrac{1}{2}(\omega_{k-1} + \omega_{k+1})\} \tag{3.43}$$

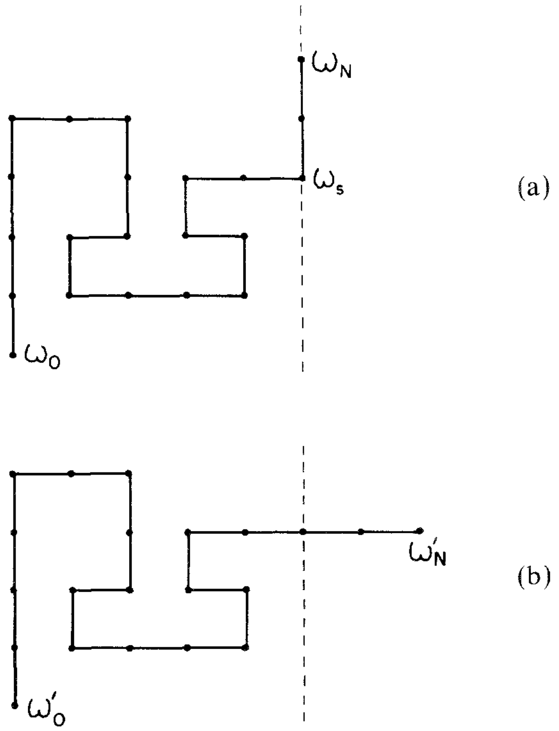

Fig. 2. (a) Rotate 90° at ω_s. (b) The result.

Thus, $(\omega_s, \omega_{s+1},..., \omega_N)$ lie on a straight line perpendicular to the line segment joining ω_{s-1} with ω_s. Let j' and j'' be the (unique). coordinates satisfying $X_{j'}(\omega_s) \neq X_{j'}(\omega_N)$ and $X_{j''}(\omega_{s-1}) \neq X_{j''}(\omega_s)$; note that $j' \neq j''$. Now perform a 90° rotation (or diagonal reflection) at ω_s to get a new SAW ω' with $\omega'_k = \omega_k$ for $k \leqslant s$, and $(\omega'_{s-1}, \omega'_s,..., \omega'_N)$ all on one straight line.

It is clear that ω' is a SAW, since $\omega'_{s+1}, \omega'_{s+2},..., \omega'_N \notin B(\omega)$. [In detail: let $i'' = 1$ or 2 be such that

$$X_{j''}(\omega_s) = m_{j''}^{i''}(\omega)$$

Then

$$X_{j''}(\omega'_{s\pm1}) = m_{j''}^{i''}(\omega) \pm (-1)^{i''}$$

i.e., ω'_{s-1} and ω'_{s+1} lie on opposite sides of the hyperplane

$$x_{j''} = m_{j''}^{i''}(\omega)$$

Since $(\omega'_s,..., \omega'_N)$ lie on a straight line, the claim follows.] It also follows from the above that

$$M_{j''}(\omega') = M_{j''}(\omega) + (N-s)$$

Also, it is easy to see that $M_{j'}(\omega') \geqslant M_{j'}(\omega) - (N-s)$, and that $M_j(\omega') = M_j(\omega)$ for all other $j \in \{1,..., d\}$. Therefore, $D(\omega') \geqslant D(\omega)$. Finally, the choice of pivot guarantees that $A(\omega') = A(\omega) + 1$.

This completes the proof of Theorem 1. ▌

We now consider other variants of the pivot algorithm, using different subsets of the lattice symmetry group for the allowed pivots. First, it is clear that either 90° rotations or diagonal reflections must be included for the algorithm to be ergodic, for otherwise $A(\omega)$ would never change; in particular, straight rods could not be transformed into anything else. However, 90° rotations *alone* are not enough (at least in $\mathbb{Z}^2$); in fact, there exists a 223-step SAW in $\mathbb{Z}^2$ that is not connected to any other SAW by 90° rotations; see Fig. 3. (We conjecture that diagonal reflections alone *do* suffice for ergodicity.) Theorem 1 shows that we do not need 180° rotations if we have axis reflections. The reverse case is the following theorem; for simplicity, we consider only $d = 2$.

Theorem 2 ($d = 2$). The pivot algorithm is ergodic for self-avoiding walks on $\mathbb{Z}^2$ provided that the 180° rotation, and either both 90° rotations or both diagonal reflections, are given nonzero probability.

Proof. We use the notation and ideas of the previous theorem. It suffices to show that any SAW ω with $A(\omega) < N-1$ can be transformed into

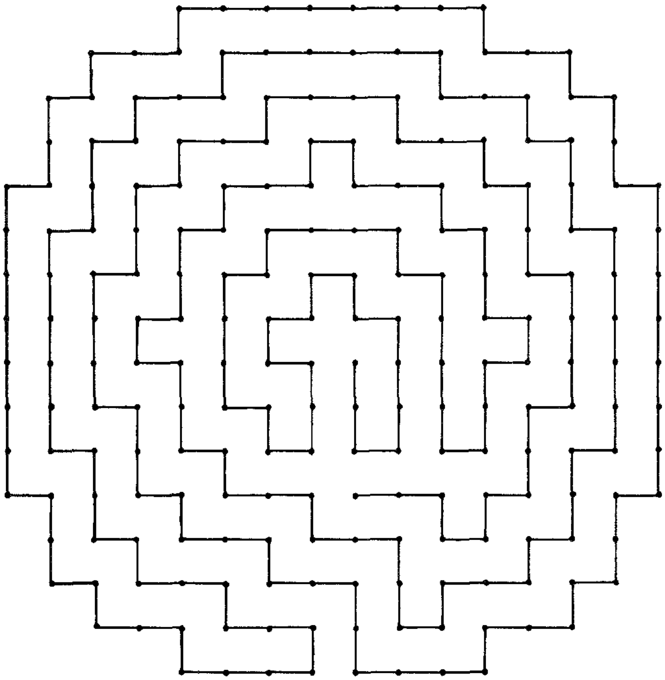

Fig. 3. A 223-step SAW in $\mathbb{Z}^2$ that is not connected to any other SAW by 90° rotations. (This SAW is not minimal.)

a SAW ω'' with $A(\omega'') = A(\omega) + 1$ by some finite sequence of allowed pivots. Let ω be an N-step SAW. Without loss of generality, assume that $X_1(\omega_{N-1}) = X_1(\omega_N)$, so that $\omega_N = \omega_{N-1} \pm (0, 1)$. If $m_1^1(\omega) = m_1^2(\omega)$, then ω is a rod (pointing in the 2-direction); so assume that $m_1^1 < m_1^2$. Choose $i \in \{1, 2\}$ so that $X_1(\omega_0) \neq m_1^i$.

Case 1. If $X_1(\omega_N) = m_1^i$, then let ω_s be the last right angle in ω, i.e.,

$$s = \max\{k: 0 < k < N \text{ and } \omega_k \neq \tfrac{1}{2}(\omega_{k-1} + \omega_{k+1})\} \tag{3.44}$$

Then a 90° rotation (or a diagonal reflection) at ω_s gives a new walk ω'' in which $(\omega_s'', ..., \omega_N'')$ lie on a straight line $x_2 = \text{const}$, and $A(\omega'') = A(\omega) + 1$. (The situation is the same as that depicted in Fig. 2.)

Case 2. If $X_1(\omega_N) \neq m_1^i$, let

$$z = \min\{x_2: (m_1^i, x_2) \in \{\omega_0, \omega_1, ..., \omega_N\}\} \tag{3.45}$$

and let t be the unique index such that $\omega_t = (m_1^i, z)$. It is not hard to see that we can rotate 180° at ω_t to get a new SAW $\bar{\omega}$ with $X_1(\bar{\omega}_{N-1}) =$

$X_1(\bar{\omega}_N)$, $A(\bar{\omega}) = A(\omega)$, and $M_1(\bar{\omega}) > M_1(\omega)$. [The inequality on M_1 is proven exactly as (3.41)–(3.42) in case (a) of Theorem 1.] We now repeat this procedure; after at most N 180° rotations we will be in case 1, and hence able to increase A by 1. ▌

Remark. This proof, unlike that of Theorem 1, shows only that the required number of pivots is at most of order N^2. We do not know if this can be improved to be of order N.

3.6. Initialization

In this section we discuss questions related to the initialization of the pivot algorithm.

When analyzing the data produced by a dynamic Monte Carlo method, one assumes that the observations come from an (approximately) stationary stochastic process whose single-time probability distribution is the desired equilibrium distribution π. This can be accomplished in either of two ways:

1. *Equilibrium start.* Choose the initial configuration X_0 from the equilibrium distribution π. Then the Markov chain X_0, X_1, X_2,... is obviously a stationary stochastic process.

2. "*Thermalization.*" Start in an arbitrary initial configuration X_0 and discard the first T observations, where T is large enough so that the distribution of X_T is very close to the stationary distribution π. Here T may be either (a) a fixed time, or (b) a stopping time.

We discuss each of these approaches in turn.

Equilibrium Start. In most applications in statistical mechanics and quantum field theory, an equilibrium start is simply unfeasible: no efficient algorithm for generating random samples from the equilibrium distribution π exists. However, the case of self-avoiding walks in the canonical (fixed-N) ensemble is an exceptionally favorable one, because there do exist feasible "static" Monte Carlo methods that choose an N-step SAW at random from the uniform distribution π. These methods are, to be sure, very time-consuming—in fact, it is an open question whether there exists such an algorithm whose expected running time is bounded by a polynomial in N. However, this is not necessarily a severe drawback to using the algorithm for the purpose of initialization, since the algorithm need only be called *once*.

The most obvious static methods are *simple sampling* and its variants: generate an ordinary random walk (or nonreversal random walk, etc.); if it intersects itself, start over; continue until you have an N-step SAW.

However, the expected running time of these methods is exponential in N, of order $(q/\mu)^N$ or $[(q-1)/\mu]^N$, respectively, where q is the coordination number of the lattice and $\mu < q-1$ is the connective constant defined in (2.1).

A much better static method is *dimerization*,[39–42] which we now explain briefly. We will then outline a heuristic argument which shows that the expected CPU time for generating an N-step SAW is

$$\sim N^{c_1 \log_2 N + c_2} \qquad \text{with} \quad c_1 = (\gamma - 1)/2$$

To generate an N-step SAW by the dimerization method, we generate two $(N/2)$-step SAWs ("dimers") and attempt to concatenate them. If the result is self-avoiding, we keep it; otherwise, we discard both dimers and try again. This algorithm is applied recursively: to generate one $(N/2)$-step SAW, we generate two $(N/4)$-step SAWs and attempt to join them (discarding both pieces if the result is not self-avoiding), etc. The recursion can stop at level k if there is a quick way to generate random SAWs of length $N_0 = 2^{-k}N$ (e.g., generating ten-step SAWs by simple sampling is quite efficient, so we can generate an 80-step SAW using three levels of recursion). It is easily proven that this algorithm generates SAWs from the uniform distribution.

Let T_N be the average amount of CPU time needed to generate an N-step SAW by dimerization. Let p_N be the probability that the concatenation of two random $(N/2)$-step SAWs yields an N-step SAW; assuming that

$$c_N \approx A\mu^N N^{\gamma - 1} \tag{3.46}$$

[cf. (2.2)], we have

$$p_N = c_N/(c_{N/2})^2 \approx B^{-1} N^{-(\gamma - 1)} \tag{3.47}$$

where $B = A/4^{\gamma - 1}$. We will need to generate, on the average, $1/p_N$ pairs of $(N/2)$-step SAWs in order to get a single N-step SAW:

$$T_N \approx BN^{\gamma - 1} 2T_{N/2} \tag{3.48}$$

(We have neglected here the time needed for checking the intersections of the two dimers; this time is linear in N, which, as will be seen shortly, is negligible compared to the time $2T_{N/2}$ for generating the two dimers.) Iterating this k times, where $k = \log_2(N/N_0)$ is the number of levels, we obtain

$$\begin{aligned} T_N &\approx \frac{(2BN^{\gamma - 1})^k}{2^{(\gamma - 1)k(k-1)/2}} T_{N_0} \\ &\sim N^{c_1 \log_2 N + c_2} \end{aligned} \tag{3.49}$$

where

$$c_1 = \frac{\gamma - 1}{2}, \qquad c_2 = \frac{\gamma + 1}{2} + \log_2 B = \frac{5 - 3\gamma}{2} + \log_2 A$$

Now, T_N grows faster than any polynomial in N, so the dimerization method will be unfeasible when N is very large. Fortunately, the constants c_1 and c_2 are very small (in $d = 2$ we have numerically $c_1 \approx 0.17$ and[11] $c_2 \approx 0.72$), so that even for N up to several hundred (resp. more than 10,000), $N^{c_1 \log_2 N + c_2}$ is less than about N^2 (resp. N^3). Thus, even for moderately large N, the large initial time investment needed to obtain exact stationarity may be feasible.

"*Thermalization.*" The usual way of dealing with initialization bias in dynamic Monte Carlo studies is to discard the first T observations, where T is chosen large enough so that the distribution of X_T is very close to the stationary distribution π; for example, $T = 10\tau_{\mathrm{exp}}$ would usually be sufficient [see the discussion surrounding (2.16)]. One then performs statistical tests to ensure that the resulting data are indeed free of initialization bias; and as an added precaution, one compares runs using radically different initial configurations X_0.

In our case we chose the initial SAW X_0 to be a straight rod. Assuming $\tau_{\mathrm{exp}} \approx N/f \sim N^{1+p}$, we get $10\tau_{\mathrm{exp}} \approx 150{,}000$ for $N = 3000$ and $10\tau_{\mathrm{exp}} \approx 600{,}000$ for $N = 10{,}000$ (see Table II in Section 4.2 for numerical data on the acceptance fraction f). These times are comparable to, but less than, our typical total run lengths. (In fact, $T = 10\tau_{\mathrm{exp}}$ may be too severe a requirement when we are primarily interested in global observables; on the other hand, $T = 10\tau_{\mathrm{int},A}$ is certainly too optimistic, since a rod is very far from equilibrium.)

For most of our runs we used the simple blanket rule $T = 200{,}000$, and carefully examined our data (for each N) to ensure that no trace of initialization bias remained. This was done by dividing the output series of 10^6 observations into 20 (or 100) "batches" of consecutive observations and looking at the means of the relevant observables within each batch. The quantities ω_N^2, S_N^2, and "f" are maximal for a rod, so the means of the first few batches should be significantly higher than the rest; and the batches corresponding to observations *after* time T should be statistically identical. An "eyeball" examination of the data (Fig. 4 in Section 4.2) shows that all visible traces of initialization bias in ω_N^2, S_N^2, and "f" have disappeared by

[11] The estimate $A \approx 1.178$ for the square lattice is easily obtained from the counts c_N in Ref. 34, using a first-order Neville–Aitken extrapolation of the sequence $c_N/\mu^N N^{\gamma-1}$ with $\mu = 2.638155$ and $\gamma = 1.34375$.

time 150,000, even at $N=10{,}000$. To make a more rigorous statistical test of the hypothesis that all initialization bias had been removed by truncation at T, we used the "combined classical and area test"[43] based on the "standardized time series" of Schruben,[44,45] which compares functionals of a transformed output series to those of a Brownian bridge. For each N, this test showed that there was an initialization bias in the complete output series, but not in the truncated series. (It should be noted, however, that this test, like many others, may not perform well if correlations of very long range (many times the batch width) are present. New statistical tests without this flaw would be a valuable asset to practitioners of Monte Carlo.) As a further check, we performed runs at $N=1000$ and 2400 with both "rod" and "dimerized" starting configurations; the results after truncation agree to within statistical error.

Finally, we remark that T need not be a fixed time, but more generally can be a *stopping time* (that is, a random time such that it can be decided whether or not $T>n$ by looking only at the observations $X_0, X_1,..., X_n$). In fact, it can be shown[46–49] for quite general Markov chains that there exists a stopping time T having the property that the distribution of X_T is *exactly* the stationary distribution π. The use of such a stopping time would bring the advantages of an "equilibrium start" (i.e., strictly zero initialization bias) without requiring a supplementary algorithm for generating samples from the equilibrium distribution π. (Indeed, the Markov chain stopped at time T serves *itself* as an algorithm for generating samples from π!) For example, in the pivot algorithm for *ordinary* random walk, we can let T be the first time such that *every* location k $(0 \leqslant k \leqslant N-1)$ has served as a pivot point. [It follows from the solution of the coupon-collector's problem[50] that the expected value of T is $N \log N + O(N)$. Note, by the way, that this is of slightly a larger order than τ_{exp} (by a logarithm). This logarithm is apparently the price one must pay for achieving a strictly zero initialization bias.] Unfortunately, we do not know of any nontrivial statistical-mechanical problem (such as the pivot algorithm for *self-avoiding* walks) in which a computationally feasible procedure for computing such a stopping time can be found.

A Hybrid Scheme. After the completion of our numerical work, we devised a "hybrid" initialization scheme that combines some of the features of dimerization and thermalization, and may have advantages over both. This scheme implements a sequence of pivot-algorithm runs at lengths $N_1 < N_2 < \cdots$. From the run at N_i (after it has attained equilibrium), one saves (e.g., on disk) a hundred or so *statistically independent* configurations of N_i-step SAWs; we return in a moment to the question of how to ensure independence. Then, to generate the initial configuration for the run at

N_{i+1}, one tries concatenating these N_i-step SAWs with independently generated $(N_{i+1}-N_i)$-step SAWs (e.g., generated by dimerization) and continues until the first success. The result is a uniformly distributed N_{i+1}-step SAW, so the run at N_{i+1} can start in equilibrium. (The runs at N_i and N_{i+1} are, to be sure, very slightly correlated. But this only causes a slight change in the *error bar* in the regression determining the critical exponents; the estimates for the exponents themselves are still unbiased.) The average number of concatenation attempts required is $c_{N_i} c_{N_{i+1}-N_i}/c_{N_{i+1}}$. For $\Delta N \equiv N_{i+1} - N_i$ fixed, this approaches $A(\Delta N)^{\gamma-1}$ as $N_i \to \infty$ (and is smaller for smaller values of N_i). For example, for $\Delta N = 1000$ and $d=2$, the average number of concatenation attempts is ≈ 13. The advantage of this initialization procedure is that it permits an equilibrium start at N_{i+1} with only minor computational overhead beyond what would have been incurred anyway (the run at N_i).

The batch of N_i-step walks must be statistically independent if the resulting N_{i+1}-step walk is to be correctly distributed. To ensure the approximate statistical independence of these walks, they should be separated by a large time interval in the pivot-algorithm sequence, e.g., $\Delta t \gtrsim 10\tau_{\text{exp}}$. Unfortunately, this requirement can be strictly fulfilled only if one makes an extremely long run at N_i, of length $\gtrsim 1000\tau_{\text{exp}} \approx 1000N/f$, and this requires a computer time roughly of order $1000N^2$. The alternative is to choose walks separated by a much smaller time Δt ($\gg \tau_{\text{int},A}$ for global observables, but $\ll \tau_{\text{exp}}$), and hope that the nonindependence of these walks does not cause too great a deviation from uniform distribution in the resulting N_{i+1}-step SAW. (The independence of this batch can be further enhanced by choosing the walk to be concatenated *randomly* from among the ones not yet chosen, rather than sequentially.) Of course, it would then be prudent to "thermalize" the pivot-algorithm run at N_{i+1} for some time T before taking data. However, since the starting configuration should be reasonably close to uniformly distributed, the needed thermalization time T should be much less than that required for a rod start. In fact, we suspect that in practice the initialization bias in the run at N_{i+1} would be undetectable.

The foregoing discussion shows that initialization can be a serious problem in the pivot algorithm, since $\tau_{\text{exp}} \gg \tau_{\text{int},A}$ for the observables of interest. Indeed, while $\tau_{\text{int},A}$ for global observables A is only of order N (in units of computer time), τ_{exp} is of order N^2, so the time T required for "thermalization" is roughly $10N^2$. The time required for a "dimerized" start is asymptotically even worse, of order $N^{c_1 \log_2 N + c_2}$. Thus, *for very large N the CPU time in the pivot algorithm will be dominated by initialization* (unless better initialization methods can be devised), and the advantages of the pivot algorithm over competing algorithms will be nullified. (The algorithms of Redner and Reynolds[11] and Berretti and Sokal[13] produce one "effectively independent" sample in a time of order N^2.) Fortunately, these difficulties appear in practice only for $N \gtrsim 10{,}000$. Indeed, for $N \lesssim 3000$, dimerization is a feasible alternative, which guarantees the complete absence of initialization bias. Moreover, the "hybrid" scheme, while not resolving any of the questions of principle (a strict implementation requires runs of length $\gtrsim 1000N^2$), should provide in practice a quick way of generating almost-equilibrium starts which require only a brief subsequent thermalization.

In our work we used both methods of dealing with initialization bias: dimerized starts for $N \leqslant 2400$, and thermalization (with a rod start) for $N \geqslant 2400$. (We do not claim that 2400 is any kind of "optimal" boundary.) In future work we hope to test the "hybrid" scheme.

4. NUMERICAL RESULTS

4.1. Preliminary Tests

In order to test our pivot-algorithm program, we generated 10^7 SAWs on the square lattice of lengths $N = 15, 20$ and compared $\langle \omega_N^2 \rangle$ and $\langle S_N^2 \rangle$ with the known exact values from direct enumeration.[34,51,52] We found

$$\begin{aligned} N = 15: \quad & \langle \omega_{15}^2 \rangle = 47.2319 \pm 0.0560 \ (47.2177) \\ & \langle S_{15}^2 \rangle = 6.7847 \pm 0.0049 \ (6.7843) \\ N = 20: \quad & \langle \omega_{20}^2 \rangle = 72.1227 \pm 0.0940 \ (72.0765) \\ & \langle S_{20}^2 \rangle = 10.2477 \pm 0.0100 \end{aligned}$$

(95% confidence limits), where the known exact values are shown in parentheses. The results agree perfectly to within statistical error (about ±0.1%).

In order to test our dimerization program, we generated 10^6 SAWs on the square lattice of length $N = 20$. We found

$$\begin{aligned} \langle \omega_{20}^2 \rangle &= 72.0755 \pm 0.0184 \ (72.0765) \\ \langle S_{20}^2 \rangle &= 10.2452 \pm 0.0070 \end{aligned}$$

(95% confidence limits). Again, the results agree perfectly with the known exact value and/or with the pivot-algorithm results to within statistical error.

Our programs used a linear-congruential pseudo-random-number generator

$$x_{n+1} = ax_n + b \pmod{m} \tag{4.1}$$

with multiplier $a = 31167285$, increment $b = 1$, and modulus $m = 2^{48}$. This generator is recommended by Knuth(53) on the basis of its excellent score on the spectral test.

4.2. Results for $\langle \omega_N^2 \rangle$, S_N^2 and Acceptance Fraction

We performed extensive Monte Carlo runs on SAWs in dimension $d = 2$ (square lattice), of lengths N ranging from 200 to 10,000. Table I shows the runs we performed and the CPU time they took; all programs were written in FORTRAN 77 and run on a Cyber 170-730 computer. The total CPU time for these runs was roughly 300 hr. We used "dimerized" starts for $N \leqslant 2400$; for larger values of N, we used "rod" starts. (In retrospect, we probably could have pushed the dimerized starts to

Table I. Summary of Pivot-Algorithm Runs on Square Lattice

N	Type of start	Number of iterations	CPU time in dimerization (sec)	CPU time in pivot algorithm	
				Total (sec)	Per iteration per N (μsec)
200	Dimer	10^6	2	4800	24.00
400	Dimer	10^6	5	8487	21.22
600	Dimer	10^6	56	11475	19.13
800	Dimer	10^6	330	15022	18.78
1000	Dimer	10^6	453	17018	17.02
1000	Dimer	8×10^6	28	133570	16.70
1000	Rod	10^7	—	169577	16.96
1200	Dimer	10^6	223	21084	17.57
1400	Dimer	10^6	397	22728	16.23
1600	Dimer	10^6	147	23679	14.80
2000	Dimer	10^6	784	29833	14.92
2400	Dimer	10^6	8835	33747	14.06
2400	Rod	10^6	—	33907	14.13
3000	Rod	10^6	—	41882	13.96
4000	Rod	10^6	—	52928	13.23
5000	Rod	10^6	—	65672	13.13
6000	Rod	10^6	—	74873	12.48
7000	Rod	10^6	—	86702	12.39
8000	Rod	983060	—	95894	12.19
8000	Rod	765595	—	NA[a]	NA[a]
9000	Rod	10^6	—	108682	12.08
10000	Rod	10^6	—	121428	12.14

[a] Datum unavailable because the authors misplaced it.

somewhat higher values of N, since the CPU times for dimerization were still negligible compared to the total run time.)

Table II shows the estimates for the mean-square end-to-end distance $\langle \omega_N^2 \rangle$, the mean-square radius of gyration $\langle S_N^2 \rangle$, and the acceptance fraction f obtained from these runs. Table III shows the acceptance fractions broken down according to symmetry-group operations. Table IV shows the estimates for the autocorrelation times $\tau_{\mathrm{int},A}$ for the observables $A = \omega_N^2$, S_N^2, ω_N, and F; here F is the observable

$$F_t = \begin{cases} 1 & \text{if the pivot at time } t \text{ is successful} \\ 0 & \text{if the pivot at time } t \text{ is not successful} \end{cases} \tag{4.2}$$

whose mean value is the acceptance fraction f. The standard deviations of these estimates are shown in parentheses; a discussion of the statistical procedures by which these error bars were determined can be found in Appendix C.

We performed least-squares regressions on these data in order to extract the critical exponents ν and p and the dynamic critical exponent q, along with the corresponding critical amplitudes. In this section we discuss the static quantities $\langle \omega_N^2 \rangle$, $\langle S_N^2 \rangle$, and f. In Section 4.3 we discuss the dynamic quantities $\tau_{\mathrm{int},A}$.

Log-log graphs of $\langle \omega_N^2 \rangle$, $\langle S_N^2 \rangle$, and f versus N are so straight that there is nothing to be gained by reproducing them here. We fit $\langle \omega_N^2 \rangle$, $\langle S_N^2 \rangle$, and f to the Ansatz AN^{power}, by performing weighted least-squares regressions of their logarithms against $\log N$, using the *a priori* error bars on the raw data points (Table II) to determine both the weights and the error bars.[54] The results are

$$\begin{aligned} \langle \omega_N^2 \rangle{:} \qquad & \nu = 0.7496 \pm 0.0008 \\ & A = 0.7764 \pm 0.0087 \\ & s^2 = 1.41 \quad (20 \text{ d.f., level} = 11\,\%) \\ \langle S_N^2 \rangle \qquad & \nu = 0.7495 \pm 0.0007 \\ & A = 0.1090 \pm 0.0011 \\ & s^2 = 1.40 \quad (20 \text{ d.f., level} = 11\,\%) \\ f{:} \qquad & p = 0.1926 \pm 0.0008 \\ & A = 0.9572 \pm 0.0052 \\ & s^2 = 1.42 \quad (20 \text{ d.f., level} = 10\,\%) \end{aligned}$$

(95% confidence intervals). Here s^2 is the weighted mean-square deviation from the regression line; it should be distributed as $1/\mathscr{D}$ times a χ^2 random

Table II. Estimates for Mean-Square and End-to-End Distance $\langle \omega_N^2 \rangle$, Mean-Square Radius of Gyration $\langle S_N^2 \rangle$, Universal Ratio $\langle S_N^2 \rangle / \langle \omega_N^2 \rangle$, and Acceptance Fraction f for Self-Avoiding Walks on Square Lattice[a]

N	Start	Iterations	Discard	$\langle \omega_N^2 \rangle$	$\langle S_N^2 \rangle$	$\langle S_N^2 \rangle / \langle \omega_N^2 \rangle$	f
200	Dimer	10^6	0	2195.2 (6.5)	308.07 (0.78)	0.14034 (0.00077)	0.34587 (0.00049)
400	Dimer	10^6	0	6153.9 (20.5)	864.35 (2.52)	0.14046 (0.00088)	0.30082 (0.00047)
600	Dimer	10^6	0	11335.5 (39.4)	1593.66 (4.96)	0.14059 (0.00093)	0.27894 (0.00046)
800	Dimer	10^6	0	17556.7 (62.9)	2454.96 (7.94)	0.13983 (0.00095)	0.26427 (0.00045)
1000	Dimer	10^6	0	24495.9 (88.3)	3428.01 (10.92)	0.13994 (0.00095)	0.25336 (0.00045)
1000	Dimer	8×10^6	0	24375.4 (31.6)	3420.55 (3.94)	0.14033 (0.00034)	0.25296 (0.00016)
1000	Rod	10^7	5×10^5	24445.6 (28.9)	3426.60 (3.62)	0.14017 (0.00031)	0.25301 (0.00014)
1200	Dimer	10^6	0	32041.0 (118.7)	4500.06 (14.82)	0.14045 (0.00098)	0.24422 (0.00044)
1400	Dimer	10^6	0	40499.4 (154.2)	5692.84 (19.38)	0.14057 (0.00101)	0.23781 (0.00043)
1600	Dimer	10^6	0	48931.2 (189.9)	6894.46 (24.26)	0.14090 (0.00104)	0.23079 (0.00043)
2000	Dimer	10^6	0	68950.8 (283.8)	9675.67 (34.62)	0.14033 (0.00108)	0.22116 (0.00043)
2400	Dimer	10^6	0	90249.4 (373.3)	12694.3 (47.5)	0.14066 (0.00111)	0.21313 (0.00041)
2400	Rod	10^6	2×10^5	90581.4 (409.7)	12783.4 (50.7)	0.14113 (0.00120)	0.21419 (0.00047)
3000	Rod	10^6	2×10^5	127220 (585)	17845.0 (72.2)	0.14027 (0.00121)	0.20493 (0.00046)
4000	Rod	10^6	2×10^5	194825 (920)	27240.4 (112.3)	0.13982 (0.00124)	0.19334 (0.00045)
5000	Rod	10^6	2×10^5	270471 (1316)	37891.4 (163.0)	0.14009 (0.00128)	0.18473 (0.00044)
6000	Rod	10^6	2×10^5	359464 (1758)	50338.2 (215.1)	0.14004 (0.00128)	0.17971 (0.00044)
7000	Rod	10^6	2×10^5	455407 (2215)	64109.9 (276.1)	0.14077 (0.00129)	0.17454 (0.00043)
8000	Rod	983060	2×10^5	552638 (2894)	77715.1 (368.7)	0.14063 (0.00140)	0.16972 (0.00043)
8000	Rod	765595	2×10^5	553027 (3259)	77352.2 (406.1)	0.13987 (0.00156)	0.16894 (0.00051)
9000	Rod	10^6	2×10^5	653760 (3371)	91858.2 (425.5)	0.14051 (0.00138)	0.16547 (0.00042)
10000	Rod	10^6	2×10^5	776717 (3987)	108383 (492)	0.13954 (0.00135)	0.16307 (0.00042)

[a] Standard deviation is shown in parentheses. "Discard" indicates number of iterations discarded for thermalization.

Table III. Estimates for Acceptance Fraction for Pivot Algorithm on Square Lattice, Broken Down by Symmetry-Group Element[a]

N	Start	Iterations	Discard	90° Rotations	Axis reflections	Diagonal reflections	180° Rotations
1000	Dimer	2×10^6	0	0.29506 (0.00061)	0.29095 (0.00060)	0.26940 (0.00059)	0.05788 (0.00044)
1000	Rod	2×10^6	2×10^5	0.29502 (0.00064)	0.29260 (0.00063)	0.26925 (0.00062)	0.05698 (0.00046)
1400	Dimer	10^6	0	0.28124 (0.00084)	0.27374 (0.00083)	0.25233 (0.00081)	0.04969 (0.00058)
1600	Dimer	10^6	0	0.27365 (0.00083)	0.26612 (0.00083)	0.24442 (0.00080)	0.04725 (0.00056)
2000	Dimer	10^6	0	0.26404 (0.00082)	0.25439 (0.00081)	0.23541 (0.00079)	0.04111 (0.00052)
2400	Dimer	10^6	0	0.25602 (0.00082)	0.24469 (0.00080)	0.22607 (0.00078)	0.03823 (0.00051)
2400	Rod	10^6	2×10^5	0.25575 (0.00091)	0.24547 (0.00090)	0.22884 (0.00088)	0.03888 (0.00057)
3000	Rod	10^6	2×10^5	0.24744 (0.00090)	0.23482 (0.00089)	0.21731 (0.00086)	0.03409 (0.00054)
4000	Rod	10^6	2×10^5	0.23437 (0.00089)	0.22200 (0.00087)	0.20551 (0.00085)	0.02957 (0.00050)
5000	Rod	10^6	2×10^5	0.22522 (0.00087)	0.21188 (0.00085)	0.19624 (0.00083)	0.02603 (0.00047)
6000	Rod	10^6	2×10^5	0.22070 (0.00087)	0.20644 (0.00085)	0.19001 (0.00082)	0.02407 (0.00045)
7000	Rod	10^6	2×10^5	0.21472 (0.00086)	0.20031 (0.00084)	0.18498 (0.00081)	0.02146 (0.00043)
8000	Rod	983060	2×10^5	0.21084 (0.00086)	0.19356 (0.00083)	0.17908 (0.00081)	0.02143 (0.00043)
8000	Rod	765595	2×10^5	0.21000 (0.00101)	0.19240 (0.00098)	0.17826 (0.00095)	0.02163 (0.00051)
9000	Rod	10^6	2×10^5	0.20611 (0.00085)	0.18917 (0.00082)	0.17402 (0.00079)	0.01908 (0.00041)
10000	Rod	10^6	2×10^5	0.20294 (0.00084)	0.18556 (0.00081)	0.17306 (0.00079)	0.01834 (0.00040)

[a] Standard deviation is shown in parentheses. "Discard" indicates number of iterations discarded for thermalization. Note that these data are available for only some of our runs.

variable with $\mathscr{D} = n - 2$ degrees of freedom (d.f.), where n is the number of data points. The observed value of s^2 thus provides a goodness-of-fit test for the assumed statistical model: an abnormally large value of s^2 would 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), while an abnormally small value of s^2 would indicate that the claimed error bars on the raw data are too large. The significance level is the probability that s^2 exceeds the observed value, assuming that the model is correct. The foregoing data are thus in good agreement with the pure power-law Ansatz and with the correctness of our raw-data error bars (perhaps the latter are slightly too small). Combining the data from $\langle \omega_N^2 \rangle$ and $\langle S_N^2 \rangle$, we obtain the estimate

$$\nu = 0.7496 \pm 0.0007 \tag{4.3}$$

Table IV. Estimates for Autocorrelation Times $\tau_{\mathrm{int},A}$ for Selected Global Observables, for Pivot Algorithm on Square Lattice[a]

N	Start	Iterations	Discard	$2\tau_{\mathrm{int},\omega_N^2}$	$2\tau_{\mathrm{int},S_N^2}$	$2\tau_{\mathrm{int},\omega_N}$	$2\tau_{\mathrm{int},F}$
200	Dimer	10^6	0	19.71 (0.39)	44.94 (1.35)	18.88 (0.37)	1.084 (0.005)
400	Dimer	10^6	0	25.04 (0.56)	58.16 (1.98)	22.51 (0.48)	1.065 (0.005)
600	Dimer	10^6	0	27.13 (0.63)	66.50 (2.43)	23.49 (0.51)	1.053 (0.005)
800	Dimer	10^6	0	28.98 (0.70)	71.56 (2.71)	26.10 (0.60)	1.055 (0.005)
1000	Dimer	10^6	0	29.27 (0.71)	68.96 (2.56)	26.51 (0.61)	1.049 (0.005)
1000	Dimer	8×10^6	0	30.26 (0.26)	72.78 (0.98)	26.67 (0.22)	1.054 (0.002)
1000	Rod	10^7	5×10^5	29.88 (0.24)	72.66 (0.90)	26.71 (0.20)	1.058 (0.002)
1200	Dimer	10^6	0	30.78 (0.76)	74.50 (2.88)	28.15 (0.67)	1.056 (0.005)
1400	Dimer	10^6	0	32.55 (0.83)	79.49 (3.17)	28.25 (0.67)	1.040 (0.005)
1600	Dimer	10^6	0	33.67 (0.88)	83.98 (3.44)	28.76 (0.69)	1.050 (0.005)
2000	Dimer	10^6	0	38.00 (1.05)	87.16 (3.64)	31.93 (0.81)	1.054 (0.005)
2400	Dimer	10^6	0	38.11 (1.05)	94.73 (4.13)	31.12 (0.78)	1.029 (0.005)
2400	Rod	10^6	2×10^5	36.24 (1.09)	84.96 (3.92)	32.32 (0.92)	1.045 (0.006)
3000	Rod	10^6	2×10^5	38.33 (1.19)	90.93 (4.34)	32.85 (0.94)	1.045 (0.006)
4000	Rod	10^6	2×10^5	39.78 (1.26)	93.54 (4.53)	34.73 (1.02)	1.028 (0.006)
5000	Rod	10^6	2×10^5	42.38 (1.38)	101.67 (5.13)	37.75 (1.16)	1.035 (0.006)
6000	Rod	10^6	2×10^5	42.59 (1.39)	99.19 (4.94)	38.77 (1.21)	1.043 (0.006)
7000	Rod	10^6	2×10^5	42.56 (1.39)	101.93 (5.15)	38.99 (1.22)	1.036 (0.006)
8000	Rod	983060	2×10^5	47.47 (1.66)	119.39 (6.59)	39.85 (1.27)	1.032 (0.006)
8000	Rod	765595	2×10^5	44.64 (1.78)	107.74 (6.65)	40.73 (1.55)	1.034 (0.007)
9000	Rod	10^6	2×10^5	47.54 (1.64)	116.76 (6.31)	43.04 (1.41)	1.029 (0.006)
10000	Rod	10^6	2×10^5	47.77 (1.65)	114.78 (6.15)	43.04 (1.41)	1.031 (0.006)

[a] Standard deviation is shown in parentheses. "Discard" indicates number of iterations discarded for thermalization.

(The estimates of $\langle \omega_N^2 \rangle$ and $\langle S_N^2 \rangle$ in each run are strongly correlated, so it would be incorrect to combine the two estimates of ν as if they were independent.)

We also tried fits in which the data point(s) from the lowest value(s) of N were discarded, as a check for the possible presence of corrections to scaling. When the point at $N = 200$ was discarded, s^2 decreased slightly (to 1.36, 1.29, and 1.24, respectively), while the exponent estimates shifted slightly (to 0.7499, 0.7499, and 0.1921). While it is tempting to prefer these new estimates for ν, which are closer to the believed exact value[33] $\nu = 3/4$, we see no valid statistical reason for doing so, in view of the negligible change in the goodness of fit.

As a further test for corrections to scaling, we fit $\langle \omega_N^2 \rangle$, $\langle S_N^2 \rangle$, and f to the Ansatz $AN^{\text{power}}(1 + C/N^{\Delta})$, for a range of *fixed* values of Δ between 0.1 and 2. In all cases the estimated correction-to-scaling amplitude C is less than 1.8 times its standard deviation, which is consistent with $C = 0$. For example, for $\Delta = 1$ we obtain

$$\begin{aligned}
\langle \omega_N^2 \rangle: \qquad & \nu = 0.7503 \pm 0.0013 \\
& A = 0.7673 \pm 0.0158 \\
& C = 1.48 \pm 2.17 \\
& s^2 = 1.39 \quad (19 \text{ d.f., level} = 12\,\%) \\
\langle S_N^2 \rangle \qquad & \nu = 0.7503 \pm 0.0012 \\
& A = 0.1076 \pm 0.0020 \\
& C = 1.55 \pm 1.89 \\
& s^2 = 1.33 \quad (19 \text{ d.f., level} = 16\,\%) \\
f: \qquad & p = 0.1918 \pm 0.0013 \\
& A = 0.9511 \pm 0.0096 \\
& C = 0.79 \pm 1.05 \\
& s^2 = 1.37 \quad (19 \text{ d.f., level} = 14\,\%)
\end{aligned}$$

(95% confidence intervals). Thus, we find *no statistically significant evidence for corrections to scaling* in $\langle \omega_N^2 \rangle$, $\langle S_N^2 \rangle$, or f in the range $200 \leqslant N \leqslant 10000$.

This is not to say, of course, that corrections to scaling are absent in the two-dimensional self-avoiding walk; the point is simply that we are working at such high values of N that any corrections to scaling (whether analytic or nonanalytic) are unobservable compared to our statistical error (raw-data standard deviations of ≈ 0.1–0.5%). Since our goal here is to obtain accurate estimates for the leading exponents ν and p, free from systematic error due to corrections to scaling, the absence of significant corrections to scaling is an asset rather than a liability. However, some workers have sought to measure corrections to scaling, and even to estimate the leading correction-to-scaling exponent Δ_1, using either series-extrapolation[49,55-61] or Monte Carlo[61-64,17] methods. It seems to us that such attempts are likely to be inconclusive. As we have seen, at large N the corrections to scaling are very small. On the other hand, some workers[61,64,17] have sought to determine the leading correction-to-scaling exponent Δ_1 by studying walks of intermediate length (e.g., $10 \lesssim N \lesssim 100$), for which the corrections to scaling would be larger than for very long walks. In our opinion this procedure is *not* justified: it is true that reducing N makes the leading correction-to-scaling term $N^{-\Delta_1}$ more prominent compared to the dominant term, but it also makes the second correction-to-scaling term $N^{-\Delta_2}$ more prominent compared to the first one. Thus, this procedure does not really estimate the exponent Δ_1, which, like other critical exponents, is defined only via the limit $N \to \infty$, but rather some effective exponent Δ_{eff}, which has no intrinsic physical meaning. We do not feel qualified to pass judgment on the series-extrapolation methods, but we suspect that at currently available series lengths ($N \lesssim 27$) they are likely to fall into a similar trap.

We also computed the amplitudes in the asymptotic relations $\langle \omega_N^2 \rangle \approx A_\omega N^{2\nu}$ and $\langle S_N^2 \rangle \approx A_S N^{2\nu}$, with ν forced to be equal to the believed exact value[33] 3/4. We found $A_\omega = 0.7719 \pm 0.0010$ and $A_S = 0.10830 \pm 0.00012$ (95% confidence intervals). These estimates are consistent with those of Rapaport,[63] but are a factor of 7–10 more precise.

The ratios $Y_N \equiv \langle S_N^2 \rangle / \langle \omega_N^2 \rangle$ are believed to converge as $N \to \infty$ to a constant Y_∞ which depends only on the dimension of the lattice. Our data (Table II) show a spectacular constancy of Y_N over the range $200 \leqslant N \leqslant 10000$: each observed value lies in the narrow interval (0.1395, 0.1412) and has a standard deviation of approximately 0.001. For a more rigorous test, we performed a least-squares fit of Y_N against $B + C/N$; the result is $B = 0.14028 \pm 0.00050$, $C = 0.01 \pm 0.34$ (95% confidence intervals) with $s^2 = 0.13$ (20 d.f., level > 99.999%). This estimate of C is consistent with zero: there is no detectable variation of Y_N with N in the range $200 \leqslant N \leqslant 10000$. We therefore redid the fit assuming $C = 0$; the result is $Y_\infty = B = 0.14029 \pm 0.00033$ (95% confidence interval) with $s^2 = 0.12$ (21 d.f., level > 99.999%). The extremely low values for s^2 indicate that the true error bars on $\langle S_N^2 \rangle / \langle \omega_N^2 \rangle$ are about a factor of three smaller than those shown in Table II, as expected (see Appendix C); and the true error bar on Y_∞ is likewise about a factor of three smaller than that given above, namely $Y_\infty = 0.14029 \pm 0.00012$. This estimate can be compared with previous estimates:

Monte Carlo, square lattice, $20 \leqslant N \leqslant 600$[65]:	$Y_\infty = 0.145 \pm 0.012$
Monte Carlo, square lattice, $160 \leqslant N \leqslant 2400$[63]:	$Y_\infty = 0.14035 \pm 0.00054$
Monte Carlo, triangular lattice, $120 \leqslant N \leqslant 2400$[63]:	$Y_\infty = 0.14018 \pm 0.00028$
extrapolation from $N \leqslant 15$, square lattice[52]:	$Y_\infty = 0.140 \pm 0.001$
extrapolation from $N \leqslant 10$, triangular lattice[52]:	$Y_\infty = 0.140 \pm 0.001$

(95% confidence intervals). Our values thus agree very closely with the estimates of Rapaport[63] and Domb and Hioe,[52] but are a factor of 2–8 more precise. (In three dimensions, however, our preliminary results are less favorable to Domb and Hioe; see Section 5.1.) See also Note Added in Proof.

Finally, we analyzed the acceptance fractions for individual symmetry-group elements g. For 90° rotations, axis reflections, and diagonal reflections, log–log graphs of f versus N show no observable curvature. Least-squares fits to the pure power-law Ansatz $f = AN^{-p}$ yield

90° rotations: $p = 0.1637 \pm 0.0020$

$A = 0.916 \pm 0.015$

$s^2 = 0.85$ (14 d.f., level = 61%)

axis reflections: $p = 0.1967 \pm 0.0021$

$A = 1.136 \pm 0.019$

$s^2 = 0.98$ (14 d.f., level = 47%)

diagonal reflections: $p = 0.1953 \pm 0.0022$

$A = 1.038 \pm 0.018$

$s^2 = 1.21$ (14 d.f., level = 26%)

(95% confidence intervals). On the other hand, the data for 180° rotations show significant curvature at the lowest value of N (1000). Fits to a pure power law with and without the $N = 1000$ data points yield

all data points: $p = 0.4898 \pm 0.0091$

$A = 1.710 \pm 0.119$

$s^2 = 1.93$ (14 d.f., level = 2%)

$N = 1000$ discarded: $p = 0.5050 \pm 0.0133$

$A = 1.943 \pm 0.208$

$s^2 = 1.26$ (12 d.f., level = 24%)

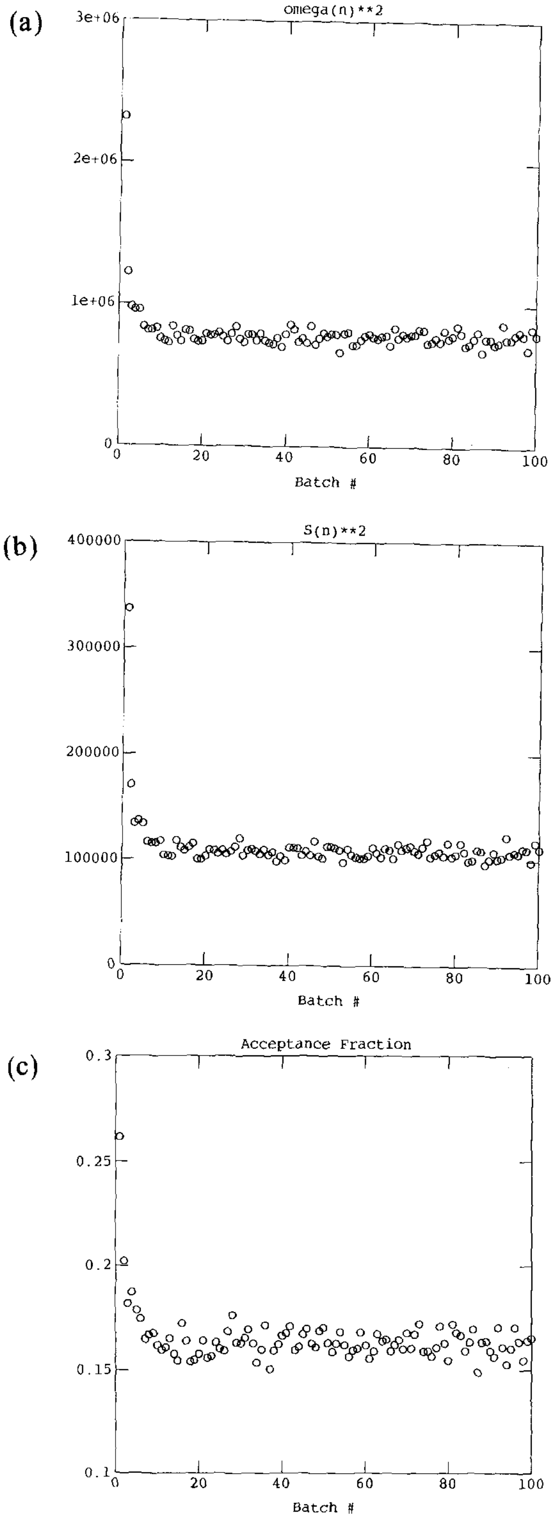

Fig. 4. Batch means of (a) ω_N^2, (b) S_N^2, and (c) f over batches of width 10,000, for pivot algorithm on square lattice at $N = 10{,}000$ with rod start.

(95% confidence intervals). Truncation to $N \geqslant 1400$ thus appears to have removed all observable corrections to scaling. Fits to the Ansatz $AN^{\text{power}}(1+C/N^{\Delta})$ for a range of fixed values of Δ confirm this assessment: if the $N=1000$ data points are discarded, the correction-to-scaling amplitude C is in all cases less than half its standard deviation, hence consistent with zero. (We find it strange that corrections to scaling should be so much stronger at $N=1000$ than at $N=1400$, but that is what the data seem to say.) All in all, for 180° rotations we estimate $p=0.505 \pm 0.03$ (we have increased the error bar because of the uncertainty regarding corrections to scaling). This estimate is not very precise, but it does resoundingly confirm the conclusion of the series analysis (Appendix A) that the acceptance-fraction exponents are *different* for different group elements g.

Table V. Data for $\langle \omega_N^2 \rangle$ and $\langle S_N^2 \rangle$[a]

$N=200$	Wall–Erpenbeck[65]	$\langle \omega_{200}^2 \rangle = 2226$ (47)	$\langle S_{200}^2 \rangle = 317$ (3.8)
	MacDonald *et al.*[16]	$\langle \omega_{200}^2 \rangle = 2201$ (19)	—
	This work (Table II)	$\langle \omega_{200}^2 \rangle = 2195.2$ (6.5)	$\langle S_{200}^2 \rangle = 308.07$ (0.78)
$N=400$	Wall–Erpenbeck[65]	$\langle \omega_{400}^2 \rangle = 6037$ (177)	$\langle S_{400}^2 \rangle = 870$ (14)
	This work (Table II)	$\langle \omega_{400}^2 \rangle = 6153.9$ (20.5)	$\langle S_{400}^2 \rangle = 864.35$ (2.52)
$N=600$	Wall–Erpenbeck[65]	$\langle \omega_{600}^2 \rangle = 9732$ (456)	$\langle S_{600}^2 \rangle = 1526$ (37)
	Mandel[66]	$\langle \omega_{600}^2 \rangle = 11470$ (149)	—
	MacDonald *et al.*[16]	$\langle \omega_{600}^2 \rangle = 11586$ (169)	—
	This work (Table II)	$\langle \omega_{600}^2 \rangle = 11335.5$ (39.4)	$\langle S_{600}^2 \rangle = 1593.66$ (4.96)
$N=800$	MacDonald *et al.*[16]	$\langle \omega_{800}^2 \rangle = 17631$ (292)	—
	This work (Table II)	$\langle \omega_{800}^2 \rangle = 17556.7$ (62.9)	$\langle S_{800}^2 \rangle = 2454.96$ (7.94)
$N=1000$	MacDonald *et al.*[16]	$\langle \omega_{1000}^2 \rangle = 25394$ (492)	—
	This work (Table II)	$\langle \omega_{1000}^2 \rangle = 24418.2$ (20.7)	$\langle S_{1000}^2 \rangle = 3424.07$ (2.59)
$N=1200$	Rapaport[63]	$\langle \omega_{1200}^2 \rangle = 31923.1$ (192)	$\langle S_{1200}^2 \rangle = 4503.13$ (18.0)
	This work (Table II)	$\langle \omega_{1200}^2 \rangle = 32041.0$ (118.7)	$\langle S_{1200}^2 \rangle = 4500.06$ (14.82)
$N=2400$	Rapaport[63]	$\langle \omega_{2400}^2 \rangle = 90963.2$ (546)	$\langle S_{2400}^2 \rangle = 12740.28$ (51)
	This work (Table II)	$\langle \omega_{2400}^2 \rangle = 90400.0$ (275.9)	$\langle S_{2400}^2 \rangle = 12735.9$ (34.7)

[a] Standard deviations are shown in parentheses. For Rapaport and Mandel, the standard deviations are those stated by the authors. For Wall and Erpenbeck and MacDonald *et al.*, who give no error bars, the standard deviations were *inferred* based on our measured values for $\langle \omega_N^2 \rangle$, $\langle \omega_N^4 \rangle$, $\langle S_N^2 \rangle$, $\langle S_N^4 \rangle$ together with the *assumption* that all their samples were independent. This assumption is probably close to correct in the case of MacDonald *et al.*, since they used a pivot algorithm and took data once every ≈ 60 iterations, which is significantly greater than our measured values for $\tau_{\text{int},\mathcal{A}}$ (see Table IV). However, in the enrichment algorithm invented and used by Wall and Erpenbeck, it is very difficult to estimate *a priori* the correlations within the batch of walks generated by a single start—so the assumption of independence may well be overoptimistic. Indeed, Rapaport, who also used the enrichment algorithm, found error bars about *twice* as large as those that would have been obtained under the assumption of independence.

As explained in Section 3.6, the initialization of the pivot algorithm is a subtle matter: for the "rod" starts, it is essential to verify that the "thermalization" interval (see the "Discard" column in Tables II–IV) is sufficiently large; otherwise the raw data would be afflicted with a severe systematic error. In Fig. 4 we plot the means of ω_N^2, S_N^2, and F over batches of width 10,000, as a function of time (batch number); this is for the run at $N = 10{,}000$, which obviously poses the most severe test. All visible traces of initialization bias have disappeared by time 150,000 at the latest. The "combined classical and area test" of Goldsman[43] confirms that there is an initialization bias in the complete output series, but not in the truncated series whenever the truncation interval is greater than 50,000.

Finally, for completeness, in Table V we compare our raw data for $\langle \omega_N^2 \rangle$ and $\langle S_N^2 \rangle$ with the results of previous workers. Wall and Erpenbeck[65] and Rapaport[63] used the enrichment algorithm,[67,10] Mandel[66] used the slithering-snake (reptation) algorithm,[68–70,66],[12] and MacDonald *et al.*[16] used the pivot algorithm. The estimates of Wall–Erpenbeck, Mandel, and MacDonald *et al.* are consistent with ours (except Wall and Erpenbeck's $\langle \omega_{600}^2 \rangle$), but they have rather large error bars. Rapaport's estimates, on the other hand, agree very closely with ours, and have error bars of the same order of magnitude. Since he and we used radically different algorithms, this agreement is compelling evidence that both his programs and ours are correct!

4.3. Results for Autocorrelation Times

We now turn to the estimates for autocorrelation times (Table IV) and attempt to estimate the dynamic critical exponent q ($\tau_{\mathrm{int},A} \sim N^q$).

Log–log plots of $\tau_{\mathrm{int},A}$ versus N show no observable curvature, except that for $A = \omega_N^2$ and S_N^2 the $N = 200$ point is anomalously low. We therefore fit $\tau_{\mathrm{int},\omega_N^2}$, $\tau_{\mathrm{int},S_N^2}$, and $\tau_{\mathrm{int},\omega_N}$ to the pure power-law Ansatz AN^q, using the methods explained above, trying fits both with and without the $N = 200$ data point. Using all data points, we find

$$\begin{aligned} \tau_{\mathrm{int},\omega_N^2}: \quad & q = 0.218 \pm 0.010 \\ & A = 3.32 \pm 0.25 \\ & s^2 = 1.97 \quad (20 \text{ d.f., level} = 0.6\,\%) \end{aligned}$$

[12] Note that the slithering-snake algorithm is nonergodic.[69–71] Thus, the Monte Carlo estimates for $\langle \omega_N^2 \rangle$ converge in the limit of infinite sample size to the mean value of ω_N^2 *over the ergodic class of a straight rod*, which is presumably slightly *larger* than the average over all SAWs. *A priori* it is difficult to estimate the magnitude of this systematic error. The close agreement between Mandel's estimate for $\langle \omega_{600}^2 \rangle$ and our own indicates that the systematic error in the slithering-snake algorithm at $N = 600$ does not exceed about 1%.

$$\begin{aligned}
\tau_{\mathrm{int},s_N^2}: \quad & q = 0.225 \pm 0.016 \\
& A = 7.58 \pm 0.88 \\
& s^2 = 1.70 \quad (20 \text{ d.f., level} = 3\,\%) \\[1ex]
\tau_{\mathrm{int},\omega_N}: \quad & q = 0.205 \pm 0.010 \\
& A = 3.23 \pm 0.23 \\
& s^2 = 0.57 \quad (20 \text{ d.f., level} = 93\,\%)
\end{aligned}$$

(95 % confidence intervals). When the point at $N = 200$ is discarded (leaving only $N \geqslant 400$), the goodness-of-fit improves dramatically, and the exponent estimates shift slightly downward:

$$\begin{aligned}
\tau_{\mathrm{int},\omega_N^2}: \quad & q = 0.206 \pm 0.012 \\
& A = 3.64 \pm 0.32 \\
& s^2 = 1.23 \quad (19 \text{ d.f., level} = 22\,\%) \\
\tau_{\mathrm{int},s_N^2}: \quad & q = 0.206 \pm 0.019 \\
& A = 8.79 \pm 1.19 \\
& s^2 = 0.84 \quad (19 \text{ d.f., level} = 66\,\%) \\
\tau_{\mathrm{int},\omega_N}: \quad & q = 0.203 \pm 0.011 \\
& A = 3.29 \pm 0.27 \\
& s^2 = 0.56 \quad (19 \text{ d.f., level} = 93\,\%)
\end{aligned}$$

(95 % confidence intervals). Discarding more low-N data points makes no further change in the fit. These latter values of q are within error bars of our estimate for p (≈ 0.193), in accordance with our heuristic argument (first half of Section 3.2) that $q = p$.

Fits to the form $\tau_{\mathrm{int},A} \sim N^q(1 + C/N^{\Delta})$ for a range of fixed values of Δ confirm the absence of significant corrections to scaling for $N \geqslant 400$: in all cases the correction-to-scaling amplitude C is less than 1.2 times its standard deviation, hence consistent with zero.

The behavior of the pivot algorithm for ordinary random walk (Section 3.3) suggests the possibility that $\tau_{\mathrm{int},A}$ might behave as $\sim N^p \log N$, where p is the acceptance-fraction exponent. To test this possibility, we fit $\tau_{\mathrm{int},A}$ to the form $\sim N^q (\log N)^D$ for a variety of *fixed* exponents D between 0 and 2. Decent fits can be obtained for all these values of D, with the leading exponent q varying from ≈ 0.20 at $D=0$ to ≈ 0.08 at $D=1$ to ≈ -0.05 at $D=2$ (this latter estimate is of course absurd). However, it seems implausible on theoretical grounds that q should be smaller than the acceptance-fraction exponent p; and indeed the only theoretical basis for considering a logarithm at all was the idea that $q=p$ with $D=1$. This latter exponent combination is strongly ruled out; and if we insist that $q \geqslant p \approx 0.19$, then D cannot be larger than about 0.2–0.3. The numerical evidence for the $d=2$ self-avoiding walk thus seems consistent with a pure power-law behavior $\tau_{\text{int},A} \sim N^q$ with $q=p$, and inconsistent with the proposed logarithmic behavior.

Remark. Our windowing procedure (see Appendix C) introduces a small downward bias in the estimates of $\tau_{\text{int},A}$ (probably a few percent). We think that the relative bias is roughly constant as a function of N. However, it is at least conceivable that the windowing algorithm introduces an N-dependent bias, which could obliterate the evidence for a logarithm if it existed. Although we do not believe that this occurred, we do want to raise the possibility. Increased confidence on this point would require a very

Table VI. Statistics on Computational Complexity of Self-Avoidance Checking, for Pivot Algorithm on Square Lattice[a]

				Total work per successful pivot				
N	Type of start	Number of iterations	$E(\text{work}\mid\text{failure})$	Failures		Success		Total
1000	Dimer	8×10^6	55.3	163.6	+	1000	=	1163.6
1000	Rod	10^7	55.0	162.7	+	1000	=	1162.7
1400	Dimer	10^6	70.3	225.4	+	1400	=	1625.4
1600	Dimer	10^6	78.6	261.9	+	1600	=	1861.9
2000	Dimer	10^6	91.0	320.4	+	2000	=	2320.4
2400	Dimer	10^6	104.6	386.3	+	2400	=	2786.3
2400	Rod	10^6	105.4	385.5	+	2400	=	2785.5
3000	Rod	10^6	123.0	475.3	+	3000	=	3475.3
4000	Rod	10^6	153.6	636.4	+	4000	=	4636.4
5000	Rod	10^6	181.7	794.0	+	5000	=	5794.0
6000	Rod	10^6	208.7	943.3	+	6000	=	6943.3
7000	Rod	10^6	234.9	1094.9	+	7000	=	8094.9
9000	Rod	10^6	285.5	1417.5	+	9000	=	10417.5
10000	Rod	10^6	309.9	1568.7	+	10000	=	11568.7

[a] Data refer to entire run, including "thermalization." (Note that these data are available for only some of our runs.) "Work" is the number of insertions into the hash table. Column 4 is the mean work per failed pivot. Column 5 is the mean work spent on failed pivots between two consecutive successful pivots.

high-precision Monte Carlo study, together with a rather sophisticated statistical analysis.

4.4. Results for Computational Complexity

We now return to the predictions made in Section 3.4 regarding the amount of work required by the pivot algorithm. The main prediction, from (3.30), is that the average amount of work spent in self-avoidance checking of failed moves *per successful pivot* should be linear in N. To test this prediction, we fit a power law AN^B to the fifth column of Table VI (using weights proportional to the length of the run); the result was $B = 0.982 \pm 0.004$, $A = 0.185 \pm 0.005$ (95% confidence intervals). [Since the raw data in Table VI lack *a priori* error bars, these confidence intervals were computed from the residual in the least-squares fit, using the Student t-test. Thus, the residual cannot be used as a *test* of the goodness of fit, as it was in the preceding cases.] This is strong support for at least the approximate linearity of the work per success as a function of N. The observed discrepancy of the exponent B from 1—more than four times the alleged error bar—should probably not be taken too seriously: it presumably reflects corrections to scaling not taken into account in the assumed power-law form. The value of A indicates that only about 15% of the work is "wasted" on failed pivots.

We also fit a power law to the "$E(\text{work} \mid \text{failure})$" column of Table VI, obtaining an exponent 0.745 ± 0.005. The predicted value, from (3.29), is $1 - p \approx 0.807$. We do not know whether this discrepancy is significant.

Finally, we fit a power law to the "CPU time per iteration per N" column of Table I, obtaining an exponent -0.173 ± 0.017. The predicted value, from (3.27), is $-p \approx -0.193$, in good agreement.

We remark that the data considered in this section come from the entire run, including the "thermalization" interval. Thus, they may not precisely measure the behavior of the pivot algorithm in equilibrium.

5. PROSPECTS FOR THE FUTURE

5.1. Three and Four Dimensions

The extension of this study to SAWs in higher dimensions is almost trivial. In any dimension, the $O(N)$ work per "effectively independent" sample should continue to hold, with a constant at worst proportional to the dimension. Runs in dimension $d = 3$ (simple cubic lattice) are now in progress; preliminary data from these runs (with $200 \leqslant N \leqslant 3000$) are shown in Table VII.[13] A least-squares regression to these preliminary data yields the estimates

$$\begin{aligned} &\langle \omega_N^2 \rangle: && \nu = 0.5907 \pm 0.0014 \\ &\langle S_N^2 \rangle && \nu = 0.5939 \pm 0.0020 \\ &f: && p = 0.1069 \pm 0.0009 \end{aligned}$$

(95% confidence intervals). These estimates for ν are in excellent agreement with the latest series-extrapolation estimate[(34)] $\nu = 0.592 \pm 0.002$ and with Rapaport's[(61)] Monte Carlo estimate $\nu = 0.592 \pm 0.004$ (95% confidence interval). Stellman and Gans[(18)] obtained a roughly similar estimate for p (≈ 0.08) in their study of the pivot algorithm for continuum polymer chains (of lengths $19 \leqslant N \leqslant 297$) in $d = 3$.

As noted previously, the ratios $Y_N = \langle S_N^2 \rangle / \langle \omega_N^2 \rangle$ are believed to converge as $N \to \infty$ to a constant Y_∞ which depends only on the dimension of the lattice. We performed a least-squares fit of Y_N against $A + B/N$; the result is $A = 0.1603 \pm 0.0017$, $B = -0.41 \pm 0.53$ (95% confidence intervals) with $s^2 = 0.18$ (7 d.f., level = 99%). This extremely low value for s^2 indicates that the true error bars on $\langle S_N^2 \rangle / \langle \omega_N^2 \rangle$ are about a factor of two smaller than those shown in Table VII, as expected (see Appendix C); and the true error bar on Y_∞ is likewise about a factor of two smaller than that given above, namely $Y_\infty = 0.1603 \pm 0.0008$. This estimate can be compared with previous estimates:

Monte Carlo, tetrahedral lattice, $40 \leqslant N \leqslant 600$[(65)]:	$Y_\infty = 0.157 \pm 0.004$
Monte Carlo, simple cubic lattice, $120 \leqslant N \leqslant 2400$[(61)]:	$Y_\infty = 0.1597 \pm 0.0006$
Monte Carlo, body-centered-cubic lattice, $120 \leqslant N \leqslant 2400$[(61)]:	$Y_\infty = 0.1594 \pm 0.0003$
extrapolation from $N \leqslant 10$, simple cubic lattice[(52)]:	$Y_\infty = 0.155 \pm 0.001$
extrapolation from $N \leqslant 8$, body-centered-cubic lattice[(52)]:	$Y_\infty = 0.155 \pm 0.001$
extrapolation from $N \leqslant 7$, face-centered cubic lattice[(52)]:	$Y_\infty = 0.155 \pm 0.001$

Our estimate of Y_∞ agrees well with Rapaport's[(61)] value, but is a factor of 2–3 less precise. However, these estimates disagree significantly with the Domb–Hioe[(52)] values. Their extrapolation predicts that Y_N decreases as N

[13] The error bars in Table VII should not be taken too seriously, as we have not yet had time to do a full time-series analysis as described in Appendix C. All error bars except those for the acceptance fraction are based on Schruben's[(45)] "combined classical sum-interval estimator." Error bars for the acceptance fraction are based on the formula for independent random variables, standard deviation $= [f(1-f)/n]^{1/2}$ (here n is the sample size), since we expect the time series $\{F_t\}$ to be essentially uncorrelated (Table II).

Table VII. Estimates for Mean-Square End-to-End Distance $\langle \omega_N^2 \rangle$, Mean-Square Radius of Gyration $\langle S_N^2 \rangle$, Universal Ratio $\langle S_N^2 \rangle / \langle \omega_N^2 \rangle$, and Acceptance Fraction f for Self-Avoiding Walks on Simple Cubic Lattice[a]

N	Start	Iterations	Discard	$\langle \omega_N^2 \rangle$	$\langle S_N^2 \rangle$	$\langle S_N^2 \rangle / \langle \omega_N^2 \rangle$	f
200	Dimer	8×10^5	0	604.06 (1.58)	95.68 (0.26)	0.15839 (0.00084)	0.56178 (0.00055)
500	Dimer	8×10^5	0	1791.90 (6.60)	284.06 (0.83)	0.15852 (0.00105)	0.50863 (0.00056)
750	Dimer	8×10^5	0	2877.69 (11.60)	459.39 (2.27)	0.15964 (0.00143)	0.48627 (0.00056)
1000	Dimer	8×10^5	0	4028.43 (18.17)	645.41 (4.84)	0.16021 (0.00192)	0.47117 (0.00056)
1250	Dimer	8×10^5	0	5288.41 (20.49)	847.93 (8.99)	0.16034 (0.00232)	0.46098 (0.00056)
1500	Dimer	8×10^5	0	6544.6 (30.6)	1047.7 (5.81)	0.16009 (0.00164)	0.45359 (0.00056)
2000	Dimer	8×10^5	0	9206.3 (40.2)	1485.2 (21.9)	0.16132 (0.00308)	0.43892 (0.00055)
2500	Dimer	8×10^5	0	11977.6 (63.6)	1921.9 (13.2)	0.16046 (0.00195)	0.42827 (0.00055)
3000	Dimer	8×10^5	0	14744.8 (63.3)	2379.3 (32.8)	0.16137 (0.00292)	0.42042 (0.00055)

[a] Standard deviation is shown in parentheses, but should be regarded as provisional.

increases (see Fig. 2 of Ref. 52), which contradicts our observations (Table VII) and those of Rapaport.[61] Clearly, extrapolation from such small values of N can be very misleading. See also Note Added in Proof.

We remark that dimerization works better in $d=3$ than in $d=2$ (due to the smaller value of $\gamma-1$), so we can go to higher N with an exact equilibrium start.

The pivot algorithm can also be used in dimension $d=4$ to look for logarithmic corrections to scaling as predicted by renormalization-group theory.[72] In particular, we hope to test the predicted logarithmic violation of hyperscaling (see Section 5.3), which is equivalent to the triviality of the corresponding (0-component!) ϕ^4 quantum field theory.[8,9]

5.2. Estimation of μ and γ

The most natural way to estimate the connective constant μ and the critical exponent γ is to use an algorithm working in a *grand canonical* (variable-N) ensemble (e.g., Redner and Reynolds[11] or Berretti and Sokal[13]). The canonical (fixed-N) ensemble, in which the pivot algorithm works, is natural for estimating ν, but rather unnatural for estimating μ and γ. [The point is that ν is defined in terms of *expectation values* in the canonical ensemble, while μ and γ involve the *partition function* (normalization factor) for this ensemble, which is not directly observable.] There do exist, however, at least two possible schemes for estimating μ and γ from the pivot-algorithm data, as we now proceed to explain.

For any N-step SAW ω, let $A_k(\omega)$ be the number of extensions of ω to an $(N+k)$-step SAW:

$$A_k(\omega) \equiv \#\{\omega' \in \mathscr{S}_k\colon\ \omega \circ \omega' \in \mathscr{S}_{N+k}\} \tag{5.1}$$

where the open dot denotes concatenation.[14] The expected value of $A_k(\omega)$, averaging over $\omega \in \mathscr{S}_N$, is c_{N+k}/c_N. Its asymptotic behavior as $N \to \infty$ at *fixed k* is

$$\frac{c_{N+k}}{c_N} \approx \mu^k \left[1 + \frac{k(\gamma-1)}{N} + o\left(\frac{1}{N}\right)\right] \tag{5.2}$$

One approach, therefore, is to take data on the observable A_k for some fixed, small value of k (e.g., $k=4$) as part of the pivot-algorithm simulation: just maintain a list of all k-step SAWs, and for each ω count how many of these k-step SAWs can be successfully appended to ω. This

[14] That is, if $\omega = (\omega_0,\ldots,\omega_N)$ and $\omega' = (\omega'_0,\ldots,\omega'_k)$ with $\omega_0 = \omega'_0 = 0$, then $\omega \circ \omega' = (\omega_0,\ldots,\omega_N, \omega_N + \omega'_1,\ldots,\omega_N + \omega'_k)$.

yields an unbiased estimate of $\langle A_k \rangle = c_{N+k}/c_N$. By doing this at several values of N and carrying out a least-squares regression to the Ansatz (5.2), one can in principle estimate μ and γ. Unfortunately, the estimates obtained in this way are likely to be mediocre for μ and even worse for γ. This can be ascertained quantitatively by working out the theory of the least-squares regression and computing the variance of the estimators $\hat{\mu}$ and $\hat{\gamma}$; we hope to do this in the near future.

An alternative approach is to run two independent pivot algorithms in parallel, at the same value of N, and attempt to concatenate the walks produced. The estimator here is the random variable

$$B(\omega^{(1)}, \omega^{(2)}) \equiv \begin{cases} 1 & \text{if} \quad \omega^{(1)} \circ \omega^{(2)} \in \mathscr{S}_{2N} \\ 0 & \text{if} \quad \omega^{(1)} \circ \omega^{(2)} \notin \mathscr{S}_{2N} \end{cases} \tag{5.3}$$

whose mean is $p_{2N} \equiv c_{2N}/c_N^2$ and whose variance is $p_{2N}(1-p_{2N})$. The asymptotic behavior of p_{2N} as $N \to \infty$ is

$$p_{2N} \equiv c_{2N}/c_N^2 \approx 2^{\gamma-1} A^{-1} N^{-(\gamma-1)} \tag{5.4}$$

where the amplitude A is defined by (3.46). Thus, by measuring $\langle B \rangle = p_{2N}$ at several values of N and carrying out a least-squares regression to the Ansatz (5.4), one can in principle estimate γ (and also A, but not μ). Again, we are unsure of the quality of these estimates; it is not hard to compute the variance of the estimator $\hat{\gamma}$, and we hope to do so in the near future.

Finally, we note that the random variables A_k and B are less "global" than the observables we have been studying in this paper (such as ω_N^2 and S_N^2), since the intersection or not of two SAWs is most strongly influenced by the behavior of those SAWs near the joining point. Thus, it is quite possible that the autocorrelation times τ_{int,A_k} and $\tau_{\text{int},B}$ in the standard pivot algorithm will have a critical exponent q that is *larger* than that found for global observables. If true, this would significantly degrade the efficiency in estimates of μ and γ. On the other hand, the probabilities p_0, $p_1,\ldots, p_{N-1}$ for choosing the pivot location are free parameters in the pivot algorithm: while we have heretofore chosen a uniform distribution $p_i = 1/N$, other distributions are permissible and might be advantageous in certain circumstances. In particular, the autocorrelation time of the observables A_k and B might be reduced by choosing these probabilities so as to focus efforts near the joining point(s), e.g., $p_i \sim (N-i)^{-\kappa}$ for some exponent $\kappa > 0$. One could then empirically determine the optimal value of κ. (On the other hand, the use of $\kappa \neq 0$ would probably increase the autocorrelation time for conventional global observables such as ω_N^2 and S_N^2. So one would either have to choose a compromise value of κ, or else perform separate runs for estimating ν and for estimating μ and γ.)

5.3. Estimation of Δ_4 and a Test of Hyperscaling

Let $\omega^{(1)}$ and $\omega^{(2)}$ be, respectively, N_1-step and N_2-step SAWs, and define $T(\omega^{(1)}, \omega^{(2)})$ to be the number of translates of $\omega^{(2)}$ that somewhere intersect $\omega^{(1)}$:

$$T(\omega^{(1)}, \omega^{(2)}) \equiv \# \{x \in \mathbb{Z}^d \colon \omega^{(1)} \cap (\omega^{(2)} + x) \neq \varnothing\} \tag{5.5a}$$

$$= \# (\omega^{(1)} - \omega^{(2)}) \tag{5.5b}$$

where $A - B \equiv \{y - z \colon y \in A, z \in B\}$. The expected value of $T(\omega^{(1)}, \omega^{(2)})$, averaging over independent walks $\omega^{(1)} \in \mathscr{S}_{N_1}$ and $\omega^{(2)} \in \mathscr{S}_{N_2}$, is $c_{N_1,N_2}/c_{N_1}c_{N_2}$. This quantity has the asymptotic behavior

$$c_{N_1,N_2}/c_{N_1}c_{N_2} \sim N_1^{2\Delta_4-\gamma} h(N_1/N_2) \tag{5.6}$$

where h is a scaling function [see (2.9)]. It is thus possible to estimate the critical exponent $2\Delta_4 - \gamma$ by running two independent pivot algorithms and measuring $T(\omega^{(1)}, \omega^{(2)})$. (Typically one would run at $N_1 = N_2 = N$ for a sequence of values of N.) In particular, this allows a direct Monte Carlo test of the hyperscaling relation $d\nu = 2\Delta_4 - \gamma$, which plays a central role in quantum field theory.[8,9] (Note that an independent measurement of γ is *not* needed.)

The efficient determination of $T(\omega^{(1)}, \omega^{(2)})$ for a specified pair of walks $(\omega^{(1)}, \omega^{(2)})$ is a very interesting and nontrivial problem in computer science. We see two broad approaches:

1. Deterministic algorithms which compute $T(\omega^{(1)}, \omega^{(2)})$ exactly.
2. Monte Carlo algorithms which produce an unbiased (or almost unbiased) estimator of $T(\omega^{(1)}, \omega^{(2)})$.

We discuss each of these approaches in turn.

Deterministic Algorithms. A straightforward method for determining $T(\omega^{(1)}, \omega^{(2)})$ is to compute $x = \omega_i^{(1)} - \omega_j^{(2)}$ for each of the $(N_1 + 1)(N_2 + 1)$ pairs (i, j), write these points x into a hash table (see Section 3.4), and count how many distinct values of x are obtained. Unfortunately, this requires a work of order $N_1 N_2$, i.e., order N^2 if $N_1 = N_2 = N$. By contrast, we expect that one "effectively independent" sample of the pair $(\omega^{(1)}, \omega^{(2)})$ can be produced in a CPU time of order N (if $\tau_{\mathrm{int},T} \sim N^p$) or in any case not much greater. So this algorithm would spend more time analyzing the data than producing it!—and the overall computational complexity per "effectively independent" sample would be increased from N to N^2, thereby nullifying the advantage of the pivot algorithm over previous[11,13] algorithms. (We remark, however, that it may be possible to devise deterministic algorithms that are more efficient than this elementary one.)

Monte Carlo Algorithms. An alternative approach is to *estimate* $T(\omega^{(1)}, \omega^{(2)})$ using an auxiliary Monte Carlo algorithm. The statistical fluctuations in this auxiliary Monte Carlo would then be added to those in the main Monte Carlo program; but this is acceptable provided that the former are comparable to or smaller than the latter.

An elegant Monte Carlo algorithm for estimating $T(\omega^{(1)}, \omega^{(2)})$ (and somewhat more general combinatorial problems) has been devised by Karp and Luby,[73] as we now briefly explain. Let $S_1,..., S_N$ be a collection of sets, each of cardinality M, and suppose we want to estimate the cardinality of $S^* \equiv \bigcup_{i=1}^{N} S_i$. [In our case, $N = N_1 + 1$, $M = N_2 + 1$, $S_i = \{\omega^{(1)}_{i-1} - \omega^{(2)}_j : j = 0,..., N_2\}$ for $1 \leqslant i \leqslant N_1 + 1$, and $T(\omega^{(1)}, \omega^{(2)}) = \#(S^*)$.] To get an unbiased estimator Z of $\#(S^*)$, we execute the following algorithm:

1. Choose $i \in \{1,..., N\}$ at random, and then choose $x \in S_i$ at random. [In our case, we take $x = \omega^{(1)}_{i-1} - \omega^{(2)}_j$, where j is chosen at random from $\{0,..., N_2\}$.] Set $t = 1$.
2. Choose $k \in \{1,..., N\}$ at random.
3. If $x \in S_k$, then go to step 4. If $x \notin S_k$, then increment t by 1 and go to step 2.
4. Put $Z = Mt$.

Then $E(Z) = T(\omega^{(1)}, \omega^{(2)})$ because

$$\begin{aligned} E(Z) &= M \sum_{l=1}^{N} E(t \mid x \text{ is in exactly } l \text{ of the sets } S_1,..., S_N) \\ &\qquad \times \operatorname{Prob}(x \text{ is in exactly } l \text{ of the sets } S_1,..., S_N) \\ &= M \sum_{l=1}^{N} \left(\frac{N}{l}\right)\left(\frac{l\ \#\{x : x \text{ is in exactly } l \text{ of the sets } S_1,..., S_N\}}{NM}\right) \\ &= \#(S^*) \end{aligned} \tag{5.7}$$

Here we used the fact that

$$\operatorname{Prob}(t = k \mid x \text{ is in exactly } l \text{ of the sets } S_1,..., S_N) = \frac{l}{N}\left(1 - \frac{l}{N}\right)^{k-1} \tag{5.8}$$

Similarly, the variance of the random variable Z can be computed; it depends on the overlap structure of the sets $S_1,..., S_N$, but lower and upper bounds are

$$\begin{aligned} \#(S^*)[\#(S^*) - M] &\leqslant \operatorname{var}(Z) \\ &\leqslant [2MN - \#(S^*)][\#(S^*) - M] && (5.9a) \\ &\leqslant 2MN\ \#(S^*) && (5.9b) \end{aligned}$$

Note that, using a hash table, each step in this algorithm can be performed in a time of order 1; thus, the total run time of the algorithm is of order $t \equiv Z/M$, which on average is $\#(S^*)/M$.

The idea is now to repeat this basic algorithm R times for the same pair $(\omega^{(1)}, \omega^{(2)})$, where R is a suitably chosen number, and to estimate $T \equiv T(\omega^{(1)}, \omega^{(2)})$ by the sample mean

$$\bar{Z}_R = \frac{1}{R} \sum_{r=1}^{R} Z_r$$

There are several alternative approaches, depending on how the number R is chosen:

(a) The simplest approach is to let R be a fixed number [the same for all pairs $(\omega^{(1)}, \omega^{(2)})$], chosen so that for "typical" pairs $(\omega^{(1)}, \omega^{(2)})$ the relative standard deviation of $\bar{Z}_R$ is suitably small (say 20%). [In view of (5.9b), it is sufficient to take R of order $MN/\#(S^*) \sim N^{2-2\Delta_4+\gamma}$ $(\sim N^{2-d\nu}$ if hyperscaling holds).] Then, for some pairs $(\omega^{(1)}, \omega^{(2)})$ the number R will be too small and the variance of $\bar{Z}_R$ will be larger than desired, while for other pairs the run-time t will be too long and the variance of $\bar{Z}_R$ will be smaller than is really needed. But such pairs will hopefully be rare enough so that neither the overall variance nor the overall run-time is adversely affected.

(b) A "sequential-sampling" approach was proposed by Karp and Luby(73): here the basic algorithm is repeated until step 3 has been performed precisely CN times, where C is a chosen constant. Thus, R is a *random* time defined as the largest number such that $\sum_{r=1}^{R} Z_r \leqslant CMN$. In this case $\bar{Z}_R$ is a *biased* estimator of $T \equiv T(\omega^{(1)}, \omega^{(2)})$, because R is correlated with the data $Z_1,..., Z_R$. [The alternative estimators $\bar{Z}_{R+1}$, CMN/R, and $CMN/(R+1)$ are also biased.] However, the bias can be bounded, and it can be shown(74) that using any of these four estimators, with a suitable choice of C we can get an estimate $\hat{Z}$ satisfying

$$\operatorname{Prob}(|\hat{Z} - T|/T \geqslant \varepsilon) \leqslant \delta \tag{5.10}$$

in a CPU time of order $(N/\varepsilon^2)\log(1/\delta)$. That is, the total amount of work needed to get a "good" estimate of T is linear in N.

(c) The disadvantage (for our application) of the preceding approach is its bias: though the bias can be made very small compared to the statistical error, it must be remembered that the Karp–Luby algorithm is to be used as a subroutine within the main Monte Carlo process, of which many iterations will be performed—and performing K iterations reduces the statistical error by a factor $K^{1/2}$ while the systematic error (bias) remains unaffected. Thus, if K is large (as it will be in a high-precision study), the bias could overwhelm the statistical error. An alternative scheme, which produces a strictly unbiased estimator, is a "two-sample" (or "double sampling") procedure[75]: one carries out an initial run using method (b), in order to get a rough estimate $\hat{T}$ for T [and perhaps also a rough estimate $\hat{V}$ for $V \equiv \operatorname{var}(Z)$]; and one uses this estimate to choose the number R for a second run, for example by $R \equiv CMN/\hat{T}$ or $R \equiv C\hat{V}/\hat{T}^2$. The key fact is that only the data from the *second* stage are used in computing the final estimator $\bar{Z}_R$; and since these data $Z_1,..., Z_R$ are independent of the random variable R, it follows that $\bar{Z}_R$ is unbiased. Presumably it can be arranged so that only a small fraction of the CPU time is consumed (wasted) in the first stage. We intend to test this method if method (a) turns out not to be adequate.

Thus, if the observable $T(\omega^{(1)}, \omega^{(2)})$ is sufficiently "global" so that $\tau_{\mathrm{int},T}$ is $O(N^p)$, then arguing as in Section 3.4, we see that the pivot algorithm combined with the Karp–Luby algorithm can produce one "effectively independent" estimate of $c_{N,N}/c_N^2$ in a CPU time of order N.

5.4. New Algorithms

The moral to be drawn from the pivot algorithm is that certain types of radically nonlocal moves can lead to extraordinarily efficient Monte Carlo algorithms, if the acceptance fraction for these moves is not too small (e.g., only a small inverse power of N) and the benefit from successful moves is sufficiently great. In particular, the relatively high acceptance fraction in the pivot algorithm is due to the fact that each of the two segments of the walk to be pivoted is already known to be self-avoiding, and this fact is preserved in the pivot process; as a result, self-intersections after pivoting can come only from overlap *between* the two segments, and there is a reasonable probability ($\sim N^{-p}$) that such overlap does not occur.

This reasoning leads one to ask whether other Monte Carlo algorithms for the self-avoiding walk might be devised, based on similarly nonlocal moves. One candidate for improved algorithms is the problem of generating SAWs in the variable-N (grand canonical) *fixed*-endpoint ensemble, as is needed for estimating efficiently the critical exponent α_{sing} [see (2.3)]. The only currently available algorithm for this ensemble (BFACF)[12] has a rather long autocorrelation time. Numerical experiments[76] in $d=2$ show that $\tau_{\mathrm{int},A} \sim \langle N \rangle^{3.0 \pm 0.4}$ for the observables $A = N, N^2, N^3$. Moreover, it can be proven rigorously[77] that $\tau_{\exp} = +\infty$ at all activities $\beta \neq 0$. This latter result arises from the existence of very slow modes associated with transitions $\omega \to \omega'$ that have $A(\omega, \omega') \gg$

$\max(|\omega|, |\omega'|)$, where $A(\omega, \omega')$ is the minimum surface area spanned by the union of ω and ω'. This suggests, therefore, supplementing the BFACF algorithm with *nonlocal* moves which are specifically designed to speed up these slow modes. Caracciolo *et al.*[78] are currently studying one algorithm of this type, in which the BFACF moves are supplemented by "cut-and-paste" moves, which cut the walk into two (or more) pieces, permute and/or invert the pieces, and then reassemble them. (The "inversion" of an N-step walk ω is, by definition, the walk $\tilde{\omega}$ defined by $\tilde{\omega}_i = \omega_N - \omega_{N-i}$. Note that this operation preserves the end-to-end distance vector, i.e., $\omega_N - \omega_0 = \tilde{\omega}_N - \tilde{\omega}_0$.) A heuristic argument similar to that in Section 3.2 suggests that the acceptance fraction of such moves should behave roughly as $\sim N^{-r}$ for some small, positive exponent r. On the other hand, moves of this kind should be extremely effective in speeding up precisely those transitions that are slow in the original BFACF algorithm. An optimal combination of BFACF and "cut-and-paste" moves might, therefore, yield an algorithm with a significantly smaller dynamic critical exponent than the original BFACF algorithm. Algorithmic improvements of this kind appear to be essential for high-precision Monte Carlo studies of the critical exponent α_{sing} in the SAW.[79]

APPENDIX A. SERIES ANALYSIS OF THE ACCEPTANCE FRACTION

We computed the acceptance fractions $f(g, k, N)$, by direct enumeration, for SAWs on the square lattice ($d = 2$) with $N \leqslant 17$. The results are reported in Table VIII. The final columns of Table VIII list the integers $a(g, k, N) \equiv c_N f(g, k, N)$. Note that $a(g, k, N) = a(g, N-k, N)$, so it suffices to list the values for $1 \leqslant k \leqslant N/2$. The preceding column lists the average of $a(g, k, N)$ over k, namely

$$a(g, N) \equiv \frac{1}{N-1} \sum_{k=1}^{N-1} a(g, k, N) \tag{A.1}$$

Data for the various group elements g are reported in rows marked 90° Rotations, 180° Rotations, Axis reflections, and Diagonal reflections. The row marked Group average is an average over the seven nonidentity elements of the group. The row marked Heuristic is the heuristic estimate (3.2). We remark that these enumerations required roughly 70 hr CPU time on a VAX-11/785 computer, using a rather inefficient program and an extremely inefficient FORTRAN compiler (the UNIX f77 compiler). An earlier enumeration for $N \leqslant 14$ required slightly less than 2 hr CPU time.

Table VIII. Acceptance Numbers $a(g, k, N) = c_N f(g, k, N)$ for the Pivot Algorithm on the Square Lattice[a]

		Details of acceptances $a(g, k, N)$							
	$a(g, N)$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
$N=2$, $c_N=12$									
90° Rotations	8.000	8							
180° Rotations	8.000	8							
Axis reflections	10.000	10							
Diagonal reflections	8.000	8							
Group average	8.571	8.6							
Heuristic	9.000	9.0							
$N=3$, $c_N=36$									
90° Rotations	24.000	24							
180° Rotations	24.000	24							
Axis reflections	30.000	30							
Diagonal reflections	24.000	24							
Group average	25.714	25.7							
Heuristic	27.000	27.0							
$N=4$, $c_N=100$									
90° Rotations	64.000	64	64						
180° Rotations	56.000	56	56						
Axis reflections	76.667	78	74						
Diagonal reflections	64.000	64	64						
Group average	66.476	66.9	65.7						
Heuristic	69.444	69.4	69.4						

[a] Integers are exact; decimals are rounded.

Table continued

Table VIII. (*Continued*)

$N = 5$, $c_N = 284$					
90° Rotations	180.000	184	176		
180° Rotations	152.000	168	136		
Axis reflections	214.000	226	202		
Diagonal reflections	180.000	184	176		
Group average	185.714	193.7	177.7		
Heuristic	194.172	201.6	186.7		
$N = 6$, $c_N = 780$					
90° Rotations	481.600	496	480	456	
180° Rotations	385.600	440	376	296	
Axis reflections	563.600	610	546	506	
Diagonal reflections	483.200	496	480	464	
Group average	491.771	520.6	484.0	449.7	
Heuristic	510.914	535.6	507.0	469.4	
$N = 7$, $c_N = 2172$					
90° Rotations	1328.000	1392	1320	1272	
180° Rotations	1037.333	1256	1016	840	
Axis reflections	1547.333	1714	1518	1410	
Diagonal reflections	1333.333	1392	1320	1288	
Group average	1350.667	1464.6	1333.1	1254.3	
Heuristic	1402.251	1512.0	1384.3	1310.4	
$N = 8$, $c_N = 5916$					
90° Rotations	3547.429	3744	3584	3376	3424
180° Rotations	2668.571	3336	2760	2120	2248
Axis reflections	4096.857	4626	4098	3738	3754
Diagonal reflections	3563.429	3744	3584	3424	3440
Group average	3583.429	3937.7	3613.1	3313.7	3354.9
Heuristic	3697.380	4028.4	3739.2	3423.2	3499.9

$N = 9$, $c_N - 16268$							
90° Rotations	9664.000	10352	9792	9296	9216		
180° Rotations	7124.000	9304	7496	5880	5816		
Axis reflections	11118.000	12786	11266	10282	10138		
Diagonal reflections	9712.000	10352	9792	9424	9280		
Group average	9730.286	10897.7	9885.1	9126.3	9012.0		
Heuristic	10020.174	11183.6	10153.8	9424.8	9318.6		
$N = 10$, $c_N = 44100$							
90° Rotations	25799.111	27808	26504	24840	24824	24240	
180° Rotations	18479.111	24888	20264	15352	15464	14376	
Axis reflections	29519.333	34494	30382	27522	27118	26642	
Diagonal reflections	25923.556	27808	26520	25176	24936	24432	
Group average	25851.873	29301.1	26725.1	24346.9	24174.3	23572.0	
Heuristic	26476.384	29887.0	27394.8	24872.2	24933.5	24112.4	
$N = 11$, $c_N = 120292$							
90° Rotations	69769.600	76144	72024	67856	66936	65888	
180° Rotations	49083.200	68408	55048	42088	41016	38856	
Axis reflections	79566.000	94350	82918	75098	73402	72062	
Diagonal reflections	70112.000	76144	72048	68808	67288	66272	
Group average	69711.200	80240.6	72718.3	66516.0	65181.1	63900.0	
Heuristic	71208.107	82030.4	74123.9	67942.7	66621.4	65322.2	
$N = 12$, $c_N = 324932$							
90° Rotations	186096.000	204304	194264	181744	180120	174920	176352
180° Rotations	127903.273	183272	148328	111480	109464	99928	101992
Axis reflections	211396.182	254102	223182	201418	196794	191494	191378
Diagonal reflections	186933.818	204304	194528	184176	180912	175832	176768
Group average	185250.753	215527.4	196039.4	178022.3	175016.6	169202.9	170141.1
Heuristic	188293.483	219426.1	199510.2	180280.3	178466.5	171161.8	173538.5
$N = 13$, $c_N = 881500$							
90° Rotations	500898.667	555808	525760	493440	485488	473720	471176
180° Rotations	338704.000	499512	401576	303352	293096	269480	265208
Axis reflections	567408.000	690506	605770	546538	531910	517266	512458
Diagonal reflections	503054.667	555808	526208	500480	487688	475872	472272
Group average	497346.667	586250.9	531007.4	483466.9	471895.4	457599.4	453860.0
Heuristic	504065.479	597849.9	538302.8	489444.6	477650.8	462485.5	458659.3

Table VIII. (*Continued*)

$N=14$, $c_N=2374444$									
90° Rotations	1334785.846	1489480	1414184	1321752	1304048	1262560	1261808	1244552	
180° Rotations	884997.538	1338040	1079624	807720	782920	706728	700296	674312	
Axis reflections	1507379.846	1856242	1627402	1465514	1425726	1380282	1365930	1353746	
Diagonal reflections	1339857.231	1489480	1416984	1339888	1309944	1267448	1262496	1245664	
Group average	1321291.912	1572634.9	1428109.1	1294575.4	1266050.9	1218186.9	1211537.7	1194605.1	
Heuristic	1333070.091	1598974.6	1445939.5	1301921.0	1278454.5	1220313.3	1221802.7	1195099.9	
$N=15$, $c_N=6416596$									
90° Rotations	3581152.000	4033880	3815136	3573288	3510128	3407552	3379224	3348856	
180° Rotations	2339706.286	3627256	2913944	2187384	2101320	1901688	1851560	1794792	
Axis reflections	4034655.714	5021926	4399842	3960782	3845150	3720282	3665566	3629042	
Diagonal reflections	3593545.143	4033880	3820336	3625576	3525976	3421352	3380264	3347432	
Group average	3536916.000	4258089.7	3854938.9	3500953.7	3409118.3	3285722.9	3243095.4	3206493.1	
Heuristic	3558019.774	4334983.7	3892295.7	3519771.0	3422730.0	3287399.3	3244745.4	3204213.2	
$N=16$, $c_N=17245332$									
90° Rotations	9534648.000	10797400	10240264	9564560	9413096	9094640	9042632	8890496	8933544
180° Rotations	6123636.800	9708600	7819464	5834456	5615608	5028296	4905288	4665704	4699720
Axis reflections	10714914.000	13476966	11800378	10611178	10298874	9939174	9782306	9638834	9628290
Diagonal reflections	9562283.733	10797400	10265512	9700456	9458184	9126984	9039104	8874912	8909152
Group average	9392475.467	11407447.4	10347396.0	9369549.1	9136559.4	8764270.3	8661910.3	8496312.6	8520241.7
Heuristic	9408805.050	11587198.1	10437583.0	9371698.4	9152729.7	8705386.0	8645894.4	8416841.2	8497414.2
$N=17$, $c_N=46466676$									
90° Rotations	25520712.000	29144488	27555936	25775992	25296096	24481568	24226024	23879656	23805936
180° Rotations	16172848.000	26219784	21050488	15745368	15071000	13512552	13051736	12436840	12295016
Axis reflections	28621332.500	36343230	31807906	28592350	27714826	26736330	26250054	25837442	25688522
Diagonal reflections	25584238.000	29144488	27606280	26164472	25415296	24578488	24213672	23831680	23719528
Group average	25089344.714	30783456.6	27855818.9	25258713.7	24560490.9	23586474.9	23204462.3	22790628.0	22674712.6
Heuristic	25060106.449	31300527.9	28041243.6	25259152.8	24494066.7	23397658.0	23011867.5	22541603.3	22434731.9

We analyzed these series by the ratio method with Neville–Aitken extrapolants, having first performed an Euler transformation

$$y=(1+\alpha)z/(1+\alpha z) \tag{A.2}$$

to reduce the effect of even–odd oscillations. More information on these series-extrapolation methods can be found in Refs. 80 and 81. These computations were carried out using the program NEVBARB, graciously supplied by Tony Guttmann. Our goal in these analyses is to estimate the critical exponent p for the acceptance fraction, defined by $f \sim N^{-p}$ as $N \to \infty$. Series extrapolation is, of course, a notoriously tricky and subjective business. Our exponent estimates are 95% subjective confidence intervals, but are *based solely on the internal consistency of the data from the given series,* not on their physical plausibility or compatibility with estimates based on other series or on Monte Carlo data. In order to be fair, we report the full Neville–Aitken tables and invite the reader to form his or her own estimates.

In Tables IX–XIII we report the results of an analysis of the series

$$f(g, N) \equiv \frac{1}{N-1} \sum_{k=1}^{N-1} f(g, k, N) \tag{A.3}$$

for the various group elements g. We show the first-, second-, and third-order Neville–Aitken extrapolants $p_n^{(1)}$, $p_n^{(2)}$, and $p_n^{(3)}$ for the values $\alpha=0.3$, 0.4 of the Euler-transform parameter. (For $\alpha \lesssim 0.3$ the even–odd oscillations are rather severe, and stable estimates of p cannot be made.) The bottom entry in each column is the average of the last five entries $(13 \leqslant N \leqslant 17)$, and the error bar is twice the spread of these entries. We consider this bottom entry to be a crude but objective measure of the "final estimate" to be obtained *from the given column of extrapolants alone* (i.e., without making further extrapolations), and of its "internal stability." However, it should not be taken blindly as an estimate of p, particularly if the extrapolants are monotonic: in this case, the true value of p is most likely located *beyond* the last extrapolant (as a higher-order extrapolant would no doubt reveal), and the "error bar" is likely to be a gross underestimate.

First we analyze the series for $f(g, N)$ averaged over the seven group elements g; the results are shown in Table IX. For $\alpha=0.3$ the second-order extrapolant is monotonic increasing, with weak even–odd oscillations; all we can say is that $p \gtrsim 0.172$. The third-order extrapolant is roughly stable, but with strong even–odd oscillations; we can make only the very rough estimate $p=0.18 \pm 0.05$. For $\alpha=0.4$ the even–odd oscillations are much weaker, but the convergence is slower (as expected); the second- and third-

Table IX. Neville–Aitken Table for Group-Averaged Acceptance Fraction[a]

	$\alpha = 0.3$			$\alpha = 0.4$		
n	$p_n^{(1)}$	$p_n^{(2)}$	$p_n^{(3)}$	$p_n^{(1)}$	$p_n^{(2)}$	$p_n^{(3)}$
3	0.2308			0.2857		
4	0.1067	−0.0174		0.0990	−0.0876	
5	0.0675	−0.0110	−0.0077	0.0691	0.0092	0.0576
6	0.0948	0.1767	0.3644	0.0880	0.1447	0.2803
7	0.0922	0.0819	−0.0603	0.0906	0.1008	0.0348
8	0.1039	0.1622	0.3228	0.0989	0.1408	0.2210
9	0.1082	0.1343	0.0644	0.1047	0.1394	0.1360
10	0.1146	0.1589	0.2329	0.1104	0.1503	0.1827
11	0.1188	0.1524	0.1294	0.1151	0.1528	0.1615
12	0.1231	0.1618	0.1992	0.1193	0.1573	0.1757
13	0.1266	0.1619	0.1623	0.1231	0.1603	0.1733
14	0.1299	0.1665	0.1896	0.1264	0.1634	0.1789
15	0.1328	0.1680	0.1761	0.1295	0.1659	0.1797
16	0.1355	0.1705	0.1861	0.1322	0.1682	0.1818
17	0.1380	0.1721	0.1818	0.1347	0.1701	0.1827
	0.133	0.168	0.179	0.129	0.166	0.179
	±0.023	±0.020	±0.055	±0.023	±0.020	±0.019

[a] Bottom entry in each column is average of the last five entries; error bar is twice the spread among these entries.

order extrapolants yield $p \gtrsim 0.170$ and $p \gtrsim 0.183$, respectively. Overall, a fair estimate would probably be $p = 0.18 \pm 0.04$.

Next we turn to the estimates for particular group elements g. For 90° rotations, the Neville–Aitken extrapolants (Table X) behave qualitatively very much like those for the group average, but the numerical value of p is lower; we estimate $p = 0.145 \pm 0.04$. Likewise, for axis reflections (Table XI) and diagonal reflections (Table XII), we find a very similar qualitative behavior, and estimate $p = 0.175 \pm 0.04$ and $p = 0.165 \pm 0.045$, respectively. All these group elements give roughly agreeing estimates for p, in the range ≈ 0.15–0.18. For 180° rotations (Table XIII), however, the estimates are radically different: though not well stabilized, the extrapolants suggest a much higher value for p, around 0.41.

At first we did not know what to make of this estimate. Does each group element have a *distinct* critical exponent p? On theoretical grounds, one would expect not. Moreover, all the series *except* the 180° rotations are at least consistent with having the same exponent p. But it is hard to

Table X. Neville–Aitken Table for Acceptance Fraction for 90° Rotations[a]

	$\alpha = 0.3$			$\alpha = 0.4$		
n	$p_n^{(1)}$	$p_n^{(2)}$	$p_n^{(3)}$	$p_n^{(1)}$	$p_n^{(2)}$	$p_n^{(3)}$
3	0.2308			0.2857		
4	0.0615	−0.1077		0.0571	−0.1714	
5	0.0390	−0.0061	0.0447	0.0399	0.0053	0.0936
6	0.0642	0.1398	0.2856	0.0584	0.1139	0.2226
7	0.0646	0.0660	−0.0446	0.0627	0.0800	0.0292
8	0.0750	0.1269	0.2487	0.0707	0.1105	0.1715
9	0.0792	0.1048	0.0495	0.0761	0.1089	0.1047
10	0.0851	0.1259	0.1892	0.0814	0.1183	0.1466
11	0.0890	0.1204	0.1013	0.0858	0.1206	0.1285
12	0.0929	0.1283	0.1596	0.0896	0.1245	0.1401
13	0.0961	0.1279	0.1266	0.0930	0.1268	0.1370
14	0.0991	0.1317	0.1505	0.0960	0.1292	0.1414
15	0.1017	0.1330	0.1397	0.0987	0.1312	0.1424
16	0.1041	0.1353	0.1493	0.1012	0.1332	0.1448
17	0.1062	0.1367	0.1460	0.1034	0.1349	0.1461
	0.101	0.133	0.142	0.098	0.131	0.142
	±0.020	±0.018	±0.048	±0.021	±0.016	±0.018

[a] Bottom entry in each column is average of the last five entries; error bar is twice the spread among these entries.

Table XI. Neville–Aitken Table for Acceptance Fraction for Axis Reflections[a]

	$\alpha = 0.3$			$\alpha = 0.4$		
n	$p_n^{(1)}$	$p_n^{(2)}$	$p_n^{(3)}$	$p_n^{(1)}$	$p_n^{(2)}$	$p_n^{(3)}$
3	0.2308			0.2857		
4	0.1231	0.0154		0.1143	−0.0571	
5	0.0752	−0.0204	−0.0384	0.0776	0.0041	0.0347
6	0.1071	0.2026	0.4257	0.0993	0.1645	0.3250
7	0.1023	0.0833	−0.0957	0.1009	0.1075	0.0220
8	0.1129	0.1656	0.3301	0.1082	0.1446	0.2189
9	0.1162	0.1358	0.0615	0.1130	0.1418	0.1347
10	0.1213	0.1575	0.2226	0.1177	0.1503	0.1759
11	0.1245	0.1500	0.1235	0.1214	0.1512	0.1544
12	0.1279	0.1580	0.1903	0.1247	0.1546	0.1678
13	0.1305	0.1574	0.1545	0.1276	0.1565	0.1653
14	0.1331	0.1613	0.1807	0.1302	0.1588	0.1705
15	0.1354	0.1623	0.1680	0.1326	0.1608	0.1714
16	0.1374	0.1645	0.1776	0.1347	0.1626	0.1734
17	0.1393	0.1657	0.1737	0.1367	0.1641	0.1743
	0.135	0.162	0.171	0.132	0.161	0.171
	±0.018	±0.017	±0.052	±0.018	±0.015	±0.018

[a] Bottom entry in each column is average of the last five entries; error bar is twice the spread among these entries.

Table XII. Neville–Aitken Table for Acceptance Fraction for Diagonal Reflections[a]

	$\alpha = 0.3$			$\alpha = 0.4$		
n	$p_n^{(1)}$	$p_n^{(2)}$	$p_n^{(3)}$	$p_n^{(1)}$	$p_n^{(2)}$	$p_n^{(3)}$
3	0.2308			0.2857		
4	0.0615	−0.1077		0.0571	−0.1714	
5	0.0390	−0.0061	0.0447	0.0399	0.0053	0.0936
6	0.0583	0.1163	0.2387	0.0537	0.0952	0.1852
7	0.0582	0.0576	−0.0304	0.0567	0.0688	0.0292
8	0.0700	0.1287	0.2710	0.0654	0.1087	0.1885
9	0.0753	0.1074	0.0539	0.0718	0.1104	0.1145
10	0.0823	0.1317	0.2047	0.0782	0.1225	0.1590
11	0.0873	0.1273	0.1118	0.0835	0.1265	0.1404
12	0.0924	0.1374	0.1780	0.0884	0.1322	0.1552
13	0.0966	0.1389	0.1457	0.0928	0.1363	0.1546
14	0.1006	0.1445	0.1721	0.0967	0.1405	0.1612
15	0.1041	0.1469	0.1604	0.1004	0.1439	0.1630
16	0.1074	0.1503	0.1706	0.1037	0.1470	0.1658
17	0.1104	0.1525	0.1672	0.1068	0.1497	0.1673
	0.104	0.147	0.163	0.100	0.143	0.162
	±0.028	±0.027	±0.053	±0.028	±0.027	±0.025

[a] Bottom entry in each column is average of the last five entries; error bar is twice the spread among these entries.

Table XIII. Neville–Aitken Table for Acceptance Fraction for 180° Rotations[a]

	$\alpha = 0.3$			$\alpha = 0.4$		
n	$p_n^{(1)}$	$p_n^{(2)}$	$p_n^{(3)}$	$p_n^{(1)}$	$p_n^{(2)}$	$p_n^{(3)}$
3	0.2308			0.2857		
4	0.2462	0.2615		0.2286	0.1714	
5	0.1724	0.0250	−0.0933	0.1749	0.0674	0.0154
6	0.2158	0.3461	0.6672	0.2038	0.2907	0.5140
7	0.2119	0.1963	−0.0285	0.2084	0.2269	0.1311
8	0.2338	0.3434	0.6378	0.2242	0.3030	0.4553
9	0.2428	0.2966	0.1794	0.2357	0.3047	0.3090
10	0.2556	0.3450	0.4902	0.2472	0.3275	0.3958
11	0.2648	0.3383	0.3148	0.2571	0.3363	0.3673
12	0.2741	0.3584	0.4390	0.2662	0.3482	0.3957
13	0.2822	0.3627	0.3821	0.2745	0.3571	0.3970
14	0.2898	0.3738	0.4293	0.2821	0.3658	0.4096
15	0.2967	0.3789	0.4067	0.2891	0.3730	0.4124
16	0.3030	0.3852	0.4234	0.2955	0.3792	0.4163
17	0.3088	0.3895	0.4172	0.3014	0.3844	0.4181
	0.296	0.378	0.412	0.289	0.372	0.411
	±0.053	±0.054	±0.095	±0.054	±0.055	±0.042

[a] Bottom entry in each column is average of the last five entries; error bar is twice the spread among these entries.

reconcile $p \approx 0.15$–0.18 with $p \approx 0.41$. As explained in Section 3.2, one would expect the acceptance fraction for 180° rotations to be smaller than that for other group elements; but, by standard ideas about universality, one would normally expect this to affect the amplitude and not the critical exponent. See, however, Section 4.2 for Monte Carlo data that support these surprising estimates.

We also performed an analysis of the series $a(g, N)$ defined in (A.1); this analysis yields estimates of the critical exponent $\gamma - p$. We form "biased" approximants using the current best estimate[34,82] $\mu = 2.638155 \pm 0.000004$, and we use the believed exact value[33] $\gamma = 43/32 = 1.34375$ to infer estimates for p. These estimates of p turn out to be consistent with those obtained by direct analysis of $f(g, N)$; we get $p \approx 0.15$–0.19 for all group elements except 180° rotations. However, the estimates based on $a(g, N)$ are somewhat more stable than those based on $f(g, N)$, a result we consider rather surprising, since one might expect the $a(g, N)$ and the c_N to contain irregularities which would partially cancel in forming the ratio $f(g, N) \equiv a(g, N)/c_N$. We thus consider the apparent greater stability of the estimates from $a(g, N)$ to be a fluke, which should not be taken too seriously.

APPENDIX B. BOUNDS ON THE EIGENVALUES OF THE PIVOT ALGORITHM FOR ORDINARY RANDOM WALK

In this appendix we prove lower and upper bounds on the next-to-leading eigenvalue λ_2 in the pivot algorithm for *ordinary* random walk in arbitrary dimension d. Recall that the case $d = 2$ was analyzed exactly in Section 3.3, with the result $\lambda_2 = 1 - O(1/N)$. We show here that this behavior continues to hold in general dimension d.

The lower bound $\lambda_2(P) \geqslant 1 - O(1/N)$ is a consequence of an easy variational (Rayleigh–Ritz) argument using the trial function $\mathbf{a}_1$; see (3.7).

On the other hand, let G be the group of symmetries of the lattice. Let v be a fixed vector which is the vector difference of two neighboring lattice points [e.g., in $\mathbb{Z}^d$ we can take $v = (1, 0,..., 0)$]. Then, to any ordered N-tuple $(g_1, g_2,..., g_N)$ of elements of G, we can associate a lattice walk $\omega = (\omega_0, \omega_1,..., \omega_N)$ by

$$\omega_0 = 0 \tag{B.1a}$$

$$\omega_i = \omega_{i-1} + g_1 g_2 g_3 \cdots g_i v \qquad \text{for} \quad 1 \leqslant i \leqslant N \tag{B.1b}$$

Denote this walk ω by $W(g_1,..., g_N)$. This map W from G^N to the set $\mathcal{S}_N$ of N-step walks is not one-to-one, but it is onto. {In fact, W is precisely $[2^{d-1}(d-1)!]^N$-to-one.}

Let $\omega = W(g_1,..., g_N)$ be a given walk. Suppose we choose to pivot at ω_k by the symmetry $h \in G$. The resulting walk $\omega' = (\omega'_0,..., \omega'_N)$ satisfies

$$\omega'_i = \omega_i \qquad \text{for} \quad i \leqslant k \tag{B.2a}$$

$$\omega'_i - \omega_k = h(\omega_i - \omega_k) \qquad \text{for} \quad i > k \tag{B.2b}$$

On the other hand, if we define g'_{k+1} by

$$g'_{k+1} = (g_1 g_2 \cdots g_k)^{-1} h g_1 g_2 \cdots g_k g_{k+1} \tag{B.3}$$

then the walk $\bar{\omega} \equiv W(g_1,..., g_k, g'_{k+1}, g_{k+2},..., g_N)$ satisfies

$$\bar{\omega}_i = \omega_i \qquad \text{for} \quad i \leqslant k \tag{B.4a}$$

$$\bar{\omega}_i = \bar{\omega}_{i-1} + h g_1 g_2 \cdots g_k g_{k+1} \cdots g_i v \qquad \text{for} \quad i > k \tag{B.4b}$$

Therefore, $\bar{\omega}_i - \bar{\omega}_{i-1} = h(\omega_i - \omega_{i-1})$ for all $i > k$, and so $\bar{\omega}_i - \bar{\omega}_k = h(\omega_i - \omega_k)$ for all $i > k$. It follows from (B.2) that $\omega' = \bar{\omega}$.

Thus, the random process of successive pivots induces a Markov chain $\{U_0, U_1,...\}$ with state space G^N. The transition probability from $(g_1,..., g_{k+1},..., g_N)$ to $(g_1,..., g'_{k+1},..., g_N)$ is p_h/N, where h is defined by (B.3), i.e.,

$$h = (g_1 \cdots g_k)\, g'_{k+1}\, g_{k+1}^{-1} (g_1 \cdots g_k)^{-1} \tag{B.5}$$

and p_h is the probability of choosing to pivot by symmetry operation h. Therefore, if the $\{p_g\}_{g \in G}$ satisfy

$$p_\alpha = p_{\beta\alpha\beta^{-1}} \qquad \text{for all} \quad \alpha, \beta \in G \tag{B.6}$$

(this is usually the case in practice), then the transition probability p_h/N does not depend on $\{g_i\}_{i \neq k+1}$. Thus, assuming (B.6), the transition probability matrix for this Markov chain is of the form

$$\hat{P} = \frac{1}{N} \sum_{i=1}^{N} I^{\otimes i-1} \otimes \hat{R} \otimes I^{\otimes N-i} \tag{B.7}$$

where I is the $|G| \times |G|$ identity matrix and $\hat{R}$ is a fixed $|G| \times |G|$ symmetric stochastic matrix

$$\hat{R} = \{p(g \to g')\} = \{p_{g'g^{-1}}\}$$

It follows that $\lambda_2(\hat{P}) = 1 - O(1/N)$ under assumption (B.6).

The above Markov chain $\{U_n\}$ on G^N is related to the pivot-algorithm chain $\{X_N\}$ on $\bar{\mathscr{S}}_N$ by $X_n = W(U_n)$. It follows that $\lambda_2(P) \leqslant \lambda_2(\hat{P})$:

$$\begin{aligned}\lambda_2(P) &= \sup_{g \not\equiv 0} \frac{\operatorname{cov}_\pi(g(X_0), g(X_1))}{\operatorname{var}_\pi(g(X_0))} \\ &= \sup_{g \not\equiv 0} \frac{\operatorname{cov}_{\tilde{\pi}}(g(W(U_0)), g(W(U_1)))}{\operatorname{var}_{\tilde{\pi}}(g(W(U_0)))} \\ &\leqslant \sup_{f \not\equiv 0} \frac{\operatorname{cov}_{\tilde{\pi}}(f(U_0), f(U_1))}{\operatorname{var}_{\tilde{\pi}}(f(U_0))} \\ &= \lambda_2(\hat{P}) \end{aligned} \tag{B.8}$$

where $\tilde{\pi}$ is the uniform measure on G^N.

In conclusion, we have proven [subject to the condition (B.6)] lower and upper bounds on $\lambda_2(P)$ of the form $1 - O(1/N)$, and hence lower and upper bounds on τ_{exp} of order N.

APPENDIX C. STATISTICAL METHODS

In this paper we have for the most part followed standard methods of statistical time-series analysis; for an excellent exposition, see the books of Priestley[83] and Anderson.[84] In this appendix we summarize these methods briefly.

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

$$\mu \equiv \langle A_t \rangle \tag{C.1}$$

unnormalized autocorrelation function

$$C(t) \equiv \langle A_s A_{s+t} \rangle - \mu^2 \tag{C.2}$$

normalized autocorrelation function

$$\rho(t) \equiv C(t)/C(0) \tag{C.3}$$

and integrated autocorrelation time

$$\tau_{\mathrm{int}} = \frac{1}{2} \sum_{t=-\infty}^{\infty} \rho(t) \tag{C.4}$$

Our goal is to estimate μ, $C(t)$, $\rho(t)$, and τ_{int} based on a finite (but large) sample $A_1, ..., A_n$ from this stochastic process.

The "natural" estimator of μ is the sample mean

$$\bar{A} \equiv \frac{1}{n} \sum_{i=1}^{n} A_i \tag{C.5}$$

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

$$\operatorname{var}(\bar{A}) = \frac{1}{n} \sum_{t=-(n-1)}^{n-1} \left(1 - \frac{|t|}{n}\right) C(t) \tag{C.6a}$$

$$\approx \frac{1}{n} (2\tau_{\text{int}})\, C(0) \qquad \text{for} \quad n \gg \tau \tag{C.6b}$$

[see (2.19) and what follows it for discussion]. Thus, even if we are interested only in the static quantity μ, it is necessary to estimate the dynamic quantity τ_{int} in order to determine valid error bars for μ.

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

$$\hat{C}(t) \equiv \frac{1}{n - |t|} \sum_{i=1}^{n-|t|} (A_i - \mu)(A_{i+|t|} - \mu) \tag{C.7}$$

if the mean μ is known, and

$$\hat{\hat{C}}(t) \equiv \frac{1}{n - |t|} \sum_{i=1}^{n-|t|} (A_i - \bar{A})(A_{i+|t|} - \bar{A}) \tag{C.8}$$

if the mean μ is unknown. We emphasize the conceptual distinction between the autocorrelation function $C(t)$, which for each t is a *number*, and the estimator $\hat{C}(t)$ or $\hat{\hat{C}}(t)$, which for each t is a *random variable*. As will become clear, this distinction is also of *practical* importance. $\hat{C}(t)$ is an unbiased estimator of $C(t)$, and $\hat{\hat{C}}(t)$ is almost unbiased (the bias is of order $1/n$) (Ref. 84, p. 463). Their variances and covariances are (Ref. 84, pp. 464–471, or Ref. 83, pp. 324–328)

$$\operatorname{var}(\hat{C}(t)) = \frac{1}{n} \sum_{m=-\infty}^{\infty} [C(m)^2 + C(m+t)\, C(m-t) + \kappa(t, m, m+t)] + o\left(\frac{1}{n}\right) \tag{C.9}$$

$$\operatorname{cov}(\hat{C}(t), \hat{C}(u)) = \frac{1}{n} \sum_{m=-\infty}^{\infty} [C(m)\, C(m+u-t) + C(m+u)\, C(m-t) + \kappa(t, m, m+u)] + o\left(\frac{1}{n}\right) \tag{C.10}$$

$[t, u \geqslant 0]$, where κ is the connected four-point autocorrelation function

$$\begin{aligned}\kappa(r, s, t) \equiv {} & \langle (A_i - \mu)(A_{i+r} - \mu)(A_{i+s} - \mu)(A_{i+t} - \mu) \rangle \\ & - C(r)\, C(t-s) - C(s)\, C(t-r) - C(t)\, C(s-r) \qquad \text{(C.11)}\end{aligned}$$

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

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

$$\hat{\rho}(t) \equiv \hat{C}(t)/\hat{C}(0) \qquad \text{(C.12)}$$

if the mean μ is known, and

$$\hat{\hat{\rho}}(t) \equiv \hat{\hat{C}}(t)/\hat{\hat{C}}(0) \qquad \text{(C.13)}$$

if the mean μ is unknown. The variances and covariances of $\hat{\rho}(t)$ and $\hat{\hat{\rho}}(t)$ can be computed (for large n) from (C.10); we omit the detailed formulas.

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

$$\hat{\tau}_{\text{int}} \overset{?}{\equiv} \frac{1}{2} \sum_{t=-(n-1)}^{n-1} \hat{\rho}(t) \qquad \text{(C.14)}$$

(or the analogous thing with $\hat{\hat{\rho}}$), *but this is wrong*! The estimator defined in (C.14) has a variance that does not go to zero as the sample size n goes to infinity (Ref. 83, pp. 420–431); so it is clearly a very bad estimator of τ_{int}. Roughly speaking, this is because the sample autocorrelations $\hat{\rho}(t)$ for $|t| \gg \tau$ contain much "noise" but little "signal"; and there are so many of them (order n) that the noise adds up to a total variance of order 1. (For a more detailed discussion, see Ref. 83, pp. 432–437.) The solution is to cut off the sum in (C.14) using a "window" $\lambda(t)$ that is ≈ 1 for $|t| \lesssim \tau$ but ≈ 0 for $|t| \gg \tau$:

$$\hat{\tau}_{\text{int}} \equiv \frac{1}{2} \sum_{t=-(n-1)}^{n-1} \lambda(t)\, \hat{\rho}(t) \qquad \text{(C.15)}$$

This retains most of the "signal," but discards most of the "noise." In particular, we use the rectangular window

$$\lambda(t) = \begin{cases} 1 & \text{if } |t| \leqslant M \\ 0 & \text{if } |t| > M \end{cases} \qquad \text{(C.16)}$$

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

$$\text{bias}(\hat{\tau}_{\text{int}}) = -\frac{1}{2} \sum_{|t| > M} \rho(t) + o\left(\frac{1}{n}\right) \qquad \text{(C.17)}$$

On the other hand, the variance of τ_{int} can be computed from (C.10); after some algebra, one obtains

$$\operatorname{var}(\hat{\tau}_{\text{int}}) \approx \frac{2(2M+1)}{n} \tau_{\text{int}}^2 \tag{C.18}$$

where we have made the approximation $\tau \ll M \ll n$. The choice of M is thus a tradeoff between bias and variance: the bias can be made small by taking M large enough so that $\rho(t)$ is negligible for $|t| > M$ (e.g., $M =$ a few times τ usually suffices), while the variance is kept small by taking M to be no larger than necessary, consistent with this constraint. We have found the following "automatic windowing" algorithm[76,85] to be convenient: choose M to be the smallest integer such that $M \geqslant c\tau_{\text{int}}(M)$. If $\rho(t)$ were roughly a pure exponential, then it would suffice to take $c \approx 4$ (since $e^{-4} < 2\%$). However, in our case $\rho(t)$ is expected to be very slowly decaying (see Section 3.3)—and this is in fact observed—so in this paper we have taken $c = 10$. We find that $\rho(M) \approx 0.01$–0.02, so we expect our estimates of τ_{int} to be systematically low by a few percent.

Thus, the standard deviations in Table II are given by (C.6b), with the estimated values $\hat{\tau}_{\text{int}}$ and $\hat{C}(0)$ replacing the theoretical ones. Similarly, the standard deviations in Table IV are given by (C.18), with $\hat{\tau}_{\text{int}}$ replacing τ_{int} on the right-hand side. The one exception to these rules is the column $\langle S_N^2 \rangle / \langle \omega_N^2 \rangle$ in Table II. To determine the error bar for $\langle S_N^2 \rangle / \langle \omega_N^2 \rangle$, one needs to know not only the variances of the estimates $\langle S_N^2 \rangle$ and $\langle \omega_N^2 \rangle$, but also their covariance. This covariance could in principle be determined by analyzing $((\omega_N^2)_t, (S_N^2)_t)$ as a bivariate time series, using the appropriate generalization of (C.5)–(C.18). However, we were lazy: we just set worst-case error bars on $\langle S_N^2 \rangle / \langle \omega_N^2 \rangle$ by using the triangle inequality. This amounts to assuming that the estimates $\langle S_N^2 \rangle$ and $\langle \omega_N^2 \rangle$ are perfectly anticorrelated, which is far from the truth—in fact, they are strongly *positively* correlated. Thus, the stated error bars on $\langle S_N^2 \rangle / \langle \omega_N^2 \rangle$ are likely to be several times too large, and this is in fact borne out by our regressions (Sections 4.2 and 5.1).

ACKNOWLEDGMENTS

We thank Juan Freire, Tony Guttmann, Naeem Jan, Michael Luby, and Chee Yap for helpful conversations and correspondence. Tony Guttmann deserves special thanks for helping with the series analysis reported in Appendix A.

The numerical computations reported here were performed at the

Academic Computing Facility of New York University. This research was supported in part by NSF grants DMS-8400955, DMS-8504033, and PHY-8413569.

NOTE ADDED IN PROOF

Benhamou and Mahoux[86] have computed the universal ratio $Y_\infty = \lim_{n \to \infty} \langle S_N^2 \rangle / \langle \omega_N^2 \rangle$ to second order in $\varepsilon = 4 - d$, using direct renormalization methods. They find

$$Y_\infty = \frac{1}{6}\left[1 - \frac{\varepsilon}{96} - \alpha_2 \varepsilon^2 + O(\varepsilon^3)\right]$$

where

$$\begin{aligned} \alpha_2 &= \frac{1}{9216} + \frac{5}{384}\left[\frac{2\pi}{3^{1/2}} \log\frac{4}{3} - 6\int_0^1 \frac{\log(1-u^2)}{1+3u^2}\,du\right] \\ &= 0.030628\ldots \end{aligned}$$

(The first-order term was found earlier in Refs. 87–89.) Three extrapolations of this series are *a priori* equally plausible: the [2/0] Padé approximant (i.e., naive substitution into the Taylor series), the [1/1] Padé approximant, and the [0/2] Padé approximant. For $\varepsilon = 1, 2$ these give

	[2/0]	[1/1]	[0/2]
$d = 3$	0.1598	0.1662	0.1601
$d = 2$	0.1428	0.1662	0.1457

The [2/0] and [0/2] approximants agree fairly well with our Monte Carlo estimates, but in our opinion this agreement is purely coincidental: the large coefficient of the ε^2 term (compared to the ε term) indicates that second-order perturbation theory, however we may choose to extrapolate it, is grossly unreliable for $\varepsilon \gtrsim 1/3$. This unreliability is reflected in the [1/1] Padé approximant, which has a pole at $\varepsilon \approx 0.34$ and makes grossly incorrect predictions at $\varepsilon = 1, 2$. The unfortunate conclusion is that this particular series is just too short/badly behaved to yield reliable information.

We would like to thank Juan Freire and Marvin Bishop for pointing out Ref. 86 to us.

REFERENCES

1. C. Domb, *Adv. Chem. Phys.* **15**:229 (1969).
2. D. S. McKenzie, *Phys. Rep.* **27**:35 (1976).
3. S. G. Whittington, *Adv. Chem. Phys.* **51**:1 (1982).
4. P. G. de Gennes, *Phys. Lett.* **38A**:339 (1972).
5. J. des Cloizeaux, *J. Phys. (Paris)* **36**:281 (1975).
6. M. Daoud *et al.*, *Macromolecules* **8**:804 (1975).
7. V. J. Emery, *Phys. Rev. B* **11**:239 (1975).
8. C. Aragão de Carvalho, S. Caracciolo, and J. Fröhlich, *Nucl. Phys. B* **215**[FS7]:209 (1983).
9. R. Fernández, J. Fröhlich, and A. D. Sokal, Random Walks, Critical Phenomena, and Triviality in Quantum Field Theory (Lecture Notes in Physics, Springer-Verlag, to appear).
10. F. T. Wall, S. Windwer, and P. J. Gans, in *Methods in Computational Physics*, Vol. 1, B. Alder, S. Fernbach, and M. Rotenberg, eds. (Academic Press, New York, 1963).
11. S. Redner and P. J. Reynolds, *J. Phys. A* **14**:2679 (1981).
12. B. Berg and D. Foerster, *Phys. Lett.* **106B**:323 (1981); C. Aragão de Carvalho and S. Caracciolo, *J. Phys. (Paris)* **44**:323 (1983); C. Aragão de Carvalho, S. Caracciolo, and J. Fröhlich, *Nucl. Phys. B* **215**[FS7]:209 (1983).
13. A. Berretti and A. D. Sokal, *J. Stat. Phys.* **40**:483 (1985).
14. M. Lal, *Molec. Phys.* **17**:57 (1969).
15. O. F. Olaj and K. H. Pelinka, *Makromol. Chem.* **177**:3413 (1976).
16. B. MacDonald, N. Jan, D. L. Hunter, and M. O. Steinitz, *J. Phys. A* **18**:2627 (1985).
17. D. L. Hunter, N. Jan, and B. MacDonald, *J. Phys. A* **19**:L543 (1986); K. Kelly, D. L. Hunter and N. Jan, *J. Phys. A* **20**:5029 (1987).
18. S. D. Stellman and P. J. Gans, *Macromolecules* **5**:516 (1972).
19. S. D. Stellman and P. J. Gans, *Macromolecules* **5**:720 (1972).
20. J. J. Freire and A. Horta, *J. Chem. Phys.* **65**:4049 (1976).
21. J. M. Hammersley, *Proc. Camb. Phil. Soc.* **53**:642 (1957).
22. J. M. Hammersley, *Proc. Camb. Phil. Soc.* **57**:516 (1961).
23. J. M. Hammersley and D. J. A. Welsh, *Q. J. Math. (Oxford) Ser. 2* **13**:108 (1962).
24. H. Kesten, *J. Math. Phys.* **4**:960 (1963).
25. H. Kesten, *J. Math. Phys.* **5**:1128 (1964).
26. G. Slade, *Commun. Math. Phys.* **110**:661 (1987).
27. J. G. Kemeny and J. L. Snell, *Finite Markov Chains* (Springer, New York, 1976).
28. M. Iosifescu, *Finite Markov Processes and Their Applications* (Wiley, Chichester, 1980).
29. K. L. Chung, *Markov Chains with Stationary Transition Probabilities*, 2nd ed. (Springer, New York, 1967).
30. E. Seneta, *Non-Negative Matrices and Markov Chains*, 2nd ed. (Springer, New York, 1981).
31. M. Hamermesh, *Group Theory and Its Application to Physical Problems* (Addison-Wesley, Reading, Massachusetts, 1962), Chapter 2.
32. J. Garcia de la Torre, A. Jiménez, and J. J. Freire, *Macromolecules* **15**:148 (1982).
33. B. Nienhuis, *J. Stat. Phys.* **34**:731 (1984).
34. A. J. Guttmann, *J. Phy. A* **20**:1839 (1987).
35. H. Saleur, *J. Phys. A* **19**:L807 (1986).
36. B. Duplantier, *Phys. Rev. B* **35**:5290 (1987).
37. D. E. Knuth, *The Art of Computer Programming*, Vol. 3 (Addison-Wesley, Reading, Massachusetts, 1973), Section 6.4.

38. E. Horowitz and S. Sahni, *Fundamentals of Data Structures* (Computer Science Press, Potomac, Maryland, 1976), Section 9.3.
39. K. Suzuki, *Bull. Chem. Soc. Japan* **41**:538 (1968).
40. Z. Alexandrowicz, *J. Chem. Phys.* **51**:561 (1969).
41. Z. Alexandrowicz and Y. Accad, *J. Chem. Phys.* **54**:5338 (1971).
42. N. Madras and A. D. Sokal, in preparation.
43. D. Goldsman, Ph. D. thesis, School of Operations Research and Industrial Engineering, Cornell University (1984).
44. L. Schruben, *Op. Res.* **30**:569 (1982).
45. L. Schruben, *Op. Res.* **31**:1090 (1983).
46. J. R. Baxter and R. V. Chacon, *Ill. J. Math.* **20**:467 (1976).
47. D. J. Aldous, *J. Lond. Math. Soc.* **25**:564 (1982).
48. D. Aldous, in *Séminaire de Probabilités XVII* (Lecture Notes in Mathematics No. 986, Springer-Verlag, Berlin, 1983).
49. D. Aldous and P. Diaconis, *Am. Math. Monthly* **93**:333 (1986).
50. W. Feller, *An Introduction to Probability Theory and Its Applications*, Vol. I, 3rd ed. (Wiley, New York, 1968), pp. 224–225.
51. P. Grassberger, *Z. Phys. B* **48**:255 (1982).
52. C. Domb and F. T. Hioe, *J. Chem. Phys.* **51**:1915 (1969).
53. D. E. Knuth, *The Art of Computer Programming*, Vol. 2, 2nd ed. (Addison-Wesley, Reading, Massachusetts, 1973), pp. 102–103.
54. S. D. Silvey, *Statistical Inference* (Chapman and Hall, London, 1975), Chapter 3.
55. I. Majid, Z. V. Djordjevic, and H. E. Stanley, *Phys. Rev. Lett.* **51**:1433 (1983).
56. J. Adler, *J. Phys. A* **16**:L515 (1983).
57. Z. V. Djordjevic, I. Majid, H. E. Stanley, and R. J. dos Santos, *J. Phys. A* **16**:L519 (1983).
58. V. Privman, *Physica* **123A**:428 (1984).
59. A. J. Guttmann, *J. Phys. A* **17**:455 (1984).
60. D. C. Rapaport, *J. Phys. A* **18**:L201 (1985).
61. D. C. Rapaport, *J. Phys. A* **18**:113 (1985).
62. S. Havlin and D. Ben-Avraham, *Phys. Rev. A* **27**:2759 (1983).
63. D. C. Rapaport, *J. Phys. A* **18**:L39 (1985).
64. J. W. Lyklema and K. Kremer, *Phys. Rev. B* **31**:3182 (1985).
65. F. T. Wall and J. J. Erpenbeck, *J. Chem. Phys.* **30**:637 (1959).
66. F. Mandel, *J. Chem. Phys.* **70**:3984 (1979).
67. F. T. Wall and J. J. Erpenbeck, *J. Chem. Phys.* **30**:634 (1959).
68. A. K. Kron, *Vysokomol. Soyed.* **7**:1228 (1965) [*Polymer Sci. USSR* **7**:1361 (1965)].
69. A. K. Kron *et al.*, *Molek. Biol.* **1**:576 (1967) [*Molec. Biol.* **1**:487 (1967)].
70. F. T. Wall and F. Mandel, *J. Chem. Phys.* **63**:4592 (1975).
71. N. Madras and A. D. Sokal, *J. Stat. Phys.* **47**:573 (1987).
72. E. Brézin, J.-C. LeGuillou, and J. Zinn-Justin, in *Phase Transitions and Critical Phenomena*, Vol. 6, C. Domb and M. S. Green, eds. (Academic Press, London, 1976).
73. R. M. Karp and M. Luby, in *24th Annual Symposium on Foundations of Computer Science* (IEEE, New York, 1983), pp. 56–64.
74. R. M. Karp, M. Luby, and N. Madras, Monte-Carlo Approximation Algorithms for Enumeration Problems, submitted to *J. Algorithms*.
75. A. Birnbaum and W. C. Healy, Jr., *Ann. Math. Stat.* **31**:662 (1960).
76. S. Caracciolo and A. D. Sokal, *J. Phys. A* **19**:L797 (1986).
77. A. D. Sokal and L. E. Thomas, in preparation.
78. S. Caracciolo, U. Glaus, and A. D. Sokal, in preparation.
79. S. Caracciolo and A. D. Sokal, *J. Phys. A* **20**:2569 (1987).

80. D. S. Gaunt and A. J. Guttmann, in *Phase Transitions and Critical Phenomena*, Vol. 3, C. Domb and M. S. Green, eds. (Academic Press, London, 1974).
81. A. J. Guttmann, in preparation, to appear in *Phase Transitions and Critical Phenomena*, C. Domb and J. L. Lebowitz, eds. (Academic Press, New York).
82. I. C. Enting and A. J. Guttmann, *J. Phys. A* **18**:1007 (1985).
83. M. B. Priestley, *Spectral Analysis and Time Series* (Academic Press, London, 1981).
84. T. W. Anderson, *The Statistical Analysis of Time Series* (Wiley, New York, 1971).
85. J. Goodman and A. D. Sokal, *Phys. Rev. Lett.* **56**:1015 (1986).
86. M. Benhamou and G. Mahoux, *J. Physique Lett.* **46**:L-689 (1985).
87. T. A. Witten and L. Schäfer, *J. Phys. A* **11**:1843 (1978).
88. J. des Cloizeaux, *J. Physique* **42**:635 (1981).
89. M. K. Kosmas, *J. Phys. A* **14**:2779 (1981).