Journal of Statistical Physics, Vol. 91, Nos. 5/6, 1998

New Upper Bounds for the Connective Constants of Self-Avoiding Walks

John Noonan[1]

Received December 17, 1996; final March 23, 1998

Using a novel implementation of the Goulden–Jackson method, we compute new rigorous upper bounds for the connective constants of self-avoiding walks, breaking Alm's previous records for rectangular (hypercubic) lattices. We also give the explicit generating functions for memory $\leqslant 8$. We then incorporate a numerical limit which gives bounds that are even better.

KEY WORDS: Goulden–Jackson method; upper bounds; connective constants; self-avoiding walks.

1. INTRODUCTION

One of the easiest to state, yet hardest to solve, enumeration problems is that of determining c_N, the number of N-step self-avoiding walks in a lattice. Such is the notoriety of this problem that it is mentioned in the *combinatorics* entry of *Encyclopedia Britannica.*

An N-step *walk* on the lattice is a sequence of points $x_0 = 0, x_1, \ldots, x_N$ such that x_i and x_{i+1} are *nearest neighbors.* It is a *self-avoiding walk* if all the x_i are distinct: $x_i \neq x_j$, $0 < i < j \leqslant N$.

There is an extensive literature on this subject.[1–18] An excellent exposition can be found in Madras and Slade.[15] A recent breakthrough is Hara and Slade's[13] determination of the asymptotic behavior of c_N for dimensions $d > 4$.

Even the computation of c_N for small values of N is a formidable computational problem. The current record, for the 2D square lattice, is due to Conway and Guttmann,[2] who computed it for $N \leqslant 51$.

[1] Department of Mathematics, Mount Vernon Nazarene College, Mount Vernon, Ohio 43050; e-mail: john.noonan@mission.mvnc.edu; WWW: http://www.math.temple.edu/~noonan.

0022-4715/98/0600-0871$15.00/0

Suppose our lattice is Z^d. We would like to know the number, $c_N^{(d)}$, of self-avoiding walks of length N. By examining the set of all walks that avoid retracing the most immediate step (which later will be referred to as memory $=2$) and walks that only take steps in one way along each direction, we find the trivial bounds, $d^N \leqslant c_N^{(d)} \leqslant 2d(2d-1)^{N-1}$. It[12, 15] is well known that for every lattice there exists a constant $\mu^{(d)}$, the connective constant of the lattice, such that $\lim_{n\to\infty}(c_N^{(d)})^{1/N} = \mu^{(d)}$.

2. PRELIMINARIES

Definition 1.0. Let $W^{(d)}$ denote the set of all walks on Z^d.

We will denote a forward (resp. backward) step in the x_i direction by i (resp. $-i$). The following are examples of walks. For example,

$$1, 2, -1, -1, 1 \in W^{(d)}, \qquad d \geqslant 2$$

and

$$2, 4, 3, 1, -2, 3, -4, 1, -2, -2, -1, -1, 3, -4, 3, -4, -3 \in W^{(d)}, \quad d \geqslant 4$$

Note. Throughout this paper, all walks discussed will be on the rectangular (hypercubic) lattice unless otherwise specified. As such, we will use $\mu^{(d)}$ to denote the connective constant for self-avoiding walks on Z^d.

Definition 1.1. Let $\lambda(word)$ denote the length of *word* and $wt(word) = s^{\lambda(word)}$.

Definition 1.2. If S is a set of walks, then we denote the weight of S as

$$wt(S) = \sum_{word \in S} wt(word)$$

Definition 1.3. Let $W^{(d)}i = W^{(d)} \times \{i\}$.

It is trivial to see that the number of walks of n steps in dimension d is $(2d)^n$. Here we demonstrate this fact as motivation for what is to come. The generating function for walks in dimension d is $wt(W^{(d)})$. We have

$$wt(W^{(d)}) = \sum_{\sigma \in W^{(d)}} s^{\lambda(\sigma)} \tag{1}$$

Let σ be a walk in $W^{(d)}$. Unless σ is the empty walk ($n=0$), σ will have a last step. Let the last step of σ be i, so that $\sigma=\sigma_1 i$ where σ_1 is made up of the first $n-1$ steps of σ. Then $\sigma_1 \in W^{(d)}$. Thus

$$W^{(d)} = \{[\ ]\} \cup \bigcup_{i=1}^{d} W^{(d)} i \cup \bigcup_{i=1}^{d} W^{(d)}(-i)$$

and

$$\begin{aligned} wt(W^{(d)}) &= wt([\ ]) + \sum_{i=1}^{d} [wt(W^{(d)} i) + wt(W^{(d)}(-i))] \\ &= 1 + \sum_{i=1}^{d} [2s \cdot wt(W^{(d)})] \\ &= 1 + 2ds \cdot wt(W^{(d)}) \end{aligned}$$

Thus the generating function for walks in dimension d is indeed

$$wt(W^{(d)}) = \frac{1}{1-2ds}$$

2. MISTAKES

Definition 2.0. A mistake is a walk that begins and ends at the same point and is otherwise self-avoiding.

Examples of mistakes are $1, 2, -1, -2$ and $-3, 3$. There is a close connection between the number of mistakes of a given size and the number of self-avoiding polygons of a given size. Indeed, the number of mistakes of length n (for $n>2$) is exactly $2n$ times the number of self-avoiding polygons of length n. If a walk is not self-avoiding, it will have at least one mistake and may have more mistakes, some of which may interact or overlap in some way. Furthermore, the set of walks with no mistakes is exactly the set of self-avoiding walks. This leads us to turn to Goulden and Jackson's method for counting words that avoid a prescribed set of undesirable sequences as subwords (factors).

I will now give a brief review of this powerful method as it applies to counting self-avoiding walks. A much more general discussion will be given in ref. 16.

Definition 2.1. A marked walk is a walk in which a subset of its mistakes is marked.

A marked walk may have all or none of its mistakes marked. If a walk has all of its mistakes marked then we say that it is *fully marked.* For example, all of the following are (different) marked walks;

$$a = \overline{1, 2, 1, 3, -1, -1, -2, -3}, \underline{-2, -1, 2, 1}, -1$$

$$b = 1, 2, 1, 3, -1, -1, -2, -3, \underline{-2, -1, 2, \overline{1}, -1}$$

$$b = \overline{1, 2, 1, 3, -1, -1, -2, -3}, \underline{-2, -1, 2, \overline{1}, -1}$$

(A marked mistake is indicated by either an overline or an underline, for readability. There is no distinction between an underlined and an overlined mistake.)

Definition 2.2. Let $\bar{W}^{(d)}$ denote the set of all walks in dimension d with *marked* mistakes.

Definition 2.3. For $\sigma \in \bar{W}^{(d)}$ let $\gamma(\sigma)$ denote the number of mistakes of σ that are marked.

For example, $\gamma(a) = 2$, $\gamma(b) = 2$, and $\gamma(c) = 3$.

Definition 2.4. For $\sigma \in \bar{W}^{(d)}$ let $\overline{wt}(\sigma) = (-1)^{\gamma(\sigma)} s^{\lambda(\sigma)}$.

The weights of our examples are, $\overline{wt}(a) = s^{13}$, $\overline{wt}(b) = s^{13}$ and $\overline{wt}(c) = -s^{13}$.

Definition 2.5. For a set, S, let $\overline{wt}(S) = \sum_{\sigma \in S} \overline{wt}(\sigma)$.

Lemma. The generating function for self-avoiding walks on dimension d is $\overline{wt}(\bar{W}^{(d)})$.

Suppose $\sigma \in W^{(d)}$ has no mistakes (it is self-avoiding) then σ will be found in $\bar{W}^{(d)}$ only once and will contribute $s^{\lambda(\sigma)}$ to $\overline{wt}(\bar{W}^{(d)})$.

Suppose $\sigma \in W^{(d)}$ has k mistakes. Then σ contributes $(-1)^r \binom{k}{r} s^{\lambda(\sigma)}$ to $\overline{wt}(\bar{W}^{(d)})$ for each $0 \leqslant r \leqslant k$. This contribution results from all possible markings of r mistakes of σ. Summing, we have

$$\sum_{r=0}^{k} (-1)^r \binom{k}{r} = \begin{cases} 1, & \text{if } r = 0 \\ 0, & \text{if } r > 0 \end{cases}$$

Thus, σ's contribution to $\overline{wt}(\bar{W}^{(d)})$ is 0. ∎

In the spirit of (1) we will try to compute the generating function for self-avoiding walks by partitioning the set $\bar{W}^{(d)}$. Each non-empty marked

walk in $\bar{W}^{(d)}$ either terminates with a marked mistake or doesn't. In our examples above, the last step of a is -1, and the last steps of b and c are parts of marked mistakes. We wish to retain the information we get from $\overline{wt}$ in order to compute a generating function. Hence we must truncate b and c by removing the entire terminating cluster of mistakes from each walk. We will call this cluster of mistakes an L-cluster. We use $\mathscr{L}^{(d)}$ to denote the set of all L-clusters on dimension d. Both of these are formally defined below. So we have

$$\bar{W}^{(d)} = \{[\ ]\} \cup \bigcup_{i=1}^{d} \bar{W}^{(d)} i \cup \bigcup_{i=1}^{d} \bar{W}^{(d)}(-i) \cup \bar{W}^{(d)} \mathscr{L}^{(d)}$$

and

$$\overline{wt}(\bar{W}^{(d)}) = 1 + 2\,ds\,\overline{wt}(\bar{W}^{(d)}) + \overline{wt}(\bar{W}^{(d)})\,\overline{wt}(\mathscr{L}^{(d)})$$

The generating function for self-avoiding walks on dimension d is

$$\overline{wt}(\bar{W}^{(d)}) = \frac{1}{1 - 2\,ds - \overline{wt}(\mathscr{L}^{(d)})} \tag{2}$$

So far our treatment is similar to the Lace Expansion of Brydges and Spencer (restricted to the finite-memory case), see ref. 14. Now it diverges. The Lace Expansion involves the extra step of grouping the L-clusters according to their laces, while in the Goulden-Jackson method, we group them according to their rightmost mistake.

3. L-CLUSTERS

As we have seen, if we want to compute the generating function for self-avoiding walks, we must compute $\overline{wt}(\mathscr{L}^{(d)})$. $\mathscr{L}^{(d)}$ is the set of all clusters of mistakes on dimension d.

Definition 3.1. A cluster of mistakes is a marked walk which has the following properties

(a) it is fully marked

(b) every letter contributes to at least one mistake

(c) its mistakes are fully overlapping.

In order to compute $\overline{wt}(\mathscr{L})$ we separate $\mathscr{L}$ into sets with common terminating mistakes.

Definition 3.2. Let $S[\mathfrak{m}]$ denote the set of all L-clusters which terminate with the mistake $\mathfrak{m}$.

We have

$$\overline{wt}(\mathscr{L}) = \sum_{\text{mistakes } \mathfrak{m}} \overline{wt}(S[\mathfrak{m}]) \tag{3}$$

In order to keep track of which mistakes interact in clusters, we define the prefix and suffix of a mistake.

Definition 3.3. The suffix of length k of a mistake $\mathfrak{m}$ is the last k steps of the mistake $\mathfrak{m}$.

Definition 3.4. The prefix of length k of a mistake $\mathfrak{m}$ is the first k steps of the mistake $\mathfrak{m}$.

If $\mathfrak{p}$ is a walk of length k then we will say that a particular mistake $\mathfrak{m}$ *has* the suffix (prefix) $\mathfrak{p}$ if the last (first) k steps of $\mathfrak{m}$ coincides with $\mathfrak{p}$.

If $\mathscr{C}$ is a cluster in $S[\mathfrak{m}]$ then either $\mathscr{C} = \mathfrak{m}$ or $\mathscr{C} = \mathscr{D}\mathfrak{s}$ such that $\mathfrak{s}$ is an suffix of $\mathfrak{m}$ and $\mathscr{D}$ is a cluster in $S[\mathfrak{m}']$ where $\mathfrak{m}'$ and $\mathfrak{m}$ overlap. More specifically, if $\mathfrak{m}$ is a mistake of length t and there exists a mistake, $\mathfrak{m}'$, such that the suffix of length k of $\mathfrak{m}'$ is the same as the prefix of length k of $\mathfrak{m}$ then every cluster in $S[\mathfrak{m}']$ may be appended with the suffix of length $t-k$ of $\mathfrak{m}$ to form a cluster in $S[\mathfrak{m}]$. Indeed, every cluster in $S[\mathfrak{m}]$ other than $\mathfrak{m}$ itself has a second to last mistake which overlaps $\mathfrak{m}$. Furthermore, if $\mathscr{C} \in S[\mathfrak{m}]$ and $\mathscr{D} \in S[\mathfrak{m}']$ are two such clusters where $\mathfrak{m}$ and $\mathfrak{m}'$ overlap in k steps then $\overline{wt}(\mathscr{C}) = (-1)\, s^{(\lambda(\mathfrak{m})-k)}\, \overline{wt}(\mathscr{D})$. Hence, we may write

$$\overline{wt}(S[\mathfrak{m}]) = \overline{wt}(\mathfrak{m}) + \sum_{\substack{p \\ p \text{ is a prefix of } m}} \left(\sum_{\substack{m' \\ p \text{ is a suffix of } m'}} (-1)\, s^{(\lambda(m)-\lambda(p))}\, \overline{wt}(S[\mathfrak{m}']) \right) \tag{4}$$

If we obtain such an equation for each and every mistake, we will have a system of linear equations which we can solve. The only problem we run into when studying self-avoiding walks is that the number of mistakes is infinite. We will address this problem by imposing a finite memory on our set of walks.

4. FINITE MEMORY

The set of all mistakes is infinite. As a result, it is very difficult to enumerate self-avoiding walks. Fisher and Sykes[5] suggested the study of

finite memory. However, their approach was "Naive Markovian" in the sense of ref. 16, while the present approach takes advantage of the powerful Goulden-Jackson method.

We develop a superset for the set of self-avoiding walks of n steps by imposing a finite memory on the set of walks. A walk is said to be self-avoiding with memory $= r$ if every subwalk of r steps is self-avoiding. Considering finite memory allows us to watch out for only a finite number of mistakes. Further, since the set of walks that are self-avoiding with memory $= r$ is a superset of the set of self-avoiding walks, $\mu_r^{(d)}$ (the connective constant for walks that are self-avoiding with memory $= r$ on dimension d) is an upper bound for $\mu^{(d)}$. In fact, $\mu_r^{(d)}$ converges to $\mu^{(d)}$.[15]

There are only $2d$ mistakes for walks that are self-avoiding with memory $= 2$. These are

$$\{[i, -i]\}_{\substack{i=-d \\ i\neq 0}}^{d}$$

We see that the mistake $[i, -i]$ overlaps the mistake $[-i, i]$ and no others. So the generating function for the clusters ending with $[i, -i]$ is

$$C_{[i,-i]} = -s^2 - sC_{[-i,i]}$$

This gives us a set of $2d$ linear equations in $2d$ unknowns (the $C_{[i,-i]}$). The solutions are

$$C_{[i,-i]} = -\frac{s^2}{s+1}$$

and the generating function (see (2) and (3)) for walks which are self-avoiding with memory $= 2$ is

$$F_2^{(d)} = \frac{1}{1 - 2ds + \sum_{\substack{i=-d \\ i\neq 0}}^{d} C_{[i,-i]}} = \frac{1}{1 - 2ds - 2d\dfrac{s^2}{s+1}} = \frac{s+1}{2ds - 1 - s}$$

From this we obtain our first estimate for the value of the connective constant $\mu^{(d)}$. Recall that $\mu_r^{(d)}$ is equal to the inverse of the smallest positive root of the denominator of a rational function. Thus, as expected, here $\mu_2^{(d)} = 2d - 1$ and we have rederived the obvious bound mentioned above, $\mu^{(d)} \leqslant 2d - 1$.

5. SYMMETRY AND MISTAKE CLASSES

The mistakes for a given dimension and memory can be partitioned into classes of mistakes. We say that 2 mistakes, m_1 and m_2 are similar (or in the same class) if m_1 may be transformed into m_2 by the action of a signed permutation of $\{\pm 1,..., \pm d\}$ (i.e., an isometry of Z^d). For example, the mistakes $[1, 2, -1, -2]$ and $[1, -2, -1, 2]$ are similar and $[1, 2, -1, 3, -2, -3]$ and $[3, -2, -3, 4, 2, -4]$ are similar.

As was demonstrated in the previous section, it is not necessary to keep track of every mistake in order to enumerate walks that are self-avoiding with finite memory. By exploiting the symmetries of the set of walks, we need only keep track of one from each class which will represent all in that class in the computations. It would be convenient to have a canonical form which we choose from each class.

The obvious choice for canonical mistake is the first lexicographically. Thus the mistake from the class with $-1, 2, 1, -2$ which we chose is $1, 2, -1, -2$. Notice that here we consider negative numbers to be lexicographically greater than positive numbers.

One can obtain a list of all canonical mistakes in one of two ways. If we have a list of all self-avoiding polygons of length k then we may obtain a list of all mistakes of length k and dividing them into classes, we can choose one representative from each class. Alternatively, we use an algorithm that generates all mistakes of length k which have the property that the first step in direction i is before the first step in direction j for all $0 < i < j$, and the first step in direction i is always before the first step in direction $-i$ for all $i > 0$.

Such an algorithm is easy to construct and can be found in the Maple package described at the end of this paper.

Before proceeding we must prove

Lemma. The set of all mistakes of length k which have the property that the first step in direction i is before the first step in direction j for all $0 < i < j$, and the first step in direction i is always before the first step in direction $-i$ for all $i > 0$ is equal to the set of canonical mistakes of length k. We shall call such a set of mistakes M_k.

To prove this lemma, we must prove two things. First, that each mistake in M_k is canonical and second, that every (non-canonical) mistake is equivalent to some mistake in M_k.

According to our definition, a mistake is canonical if it is the least, lexicographically in its equivalence class. If $m \in M_k$ then if we permute the directions of m in any way, we will get a mistake that is greater lexicographically than m. Indeed, any permutation (except the identity) will have

a lowest direction that it changes, call this direction j which will be replaced with another direction j', where $j' > j$.

So, M_k contains only canonical mistakes.

Suppose $\mathfrak{m}$ is a canonical mistake. Let i and j be directions that $\mathfrak{m}$ assumes with $0 < i < j$. Then the first step in direction i must precede the first step in direction j. Otherwise, one could permute i and j to form a similar mistake which is less than $\mathfrak{m}$ lexicographically. Furthermore, if $\mathfrak{m}$ has a step in the $-i$ direction before its first step in the i direction for any $i > 0$, then one could permute i and $-i$ to form a mistake that lexicographically less. Thus $\mathfrak{m} \in M_k$. ▮

6. MEMORY = 4

There are two mistake classes in memory = 4 whose canonical mistakes are $\mathfrak{m}_1 = [1, -1]$ and $\mathfrak{m}_2 = [1, 2, -1, -2]$. From these, we derive the equations

$$\begin{aligned} C_{[1,-1]} &= -s^2 - sC_{[1,-1]} - 2(d-1)\, sC_{[1,2,-1,-2]} \\ C_{[1,2,-1,-2]} &= -s^4 - sC_{[1,2,-1,-2]} - s^2 C_{[1,2,-1,-2]} \\ &\quad - s^3 C_{[1,-1]} - 2(d-1)\, s^3 C_{[1,2,-1,-2]} \end{aligned}$$

The solutions to these equations are

$$C_{[1,-1]} = \frac{-(s^2+s+1)\, s^2}{1+2s+2s^2+(2d-1)\, s^3}$$

$$C_{[1,2,-1,-2]} = \frac{-s^4}{1+2s+2s^2+(2d-1)\, s^3}$$

This leads to the following generating function for memory = 4, whose denominator was first derived by Fisher and Sykes.[5]

$$\begin{aligned} F_4^{(d)} &= \frac{1}{1-2ds-2dC_{[1,-1]}-4d(d-1)\, C_{[1,2,-1,-2]}} \\ &= \frac{1+2s+2s^2+(2d-1)\, s^3}{1-2(d-1)\, s-2(d-1)\, s^2-s^3} \end{aligned}$$

Note that the we have a system of two equations with two unknowns while Fisher and Sykes had a system of three equations with three unknowns[5] (p. 57, Eq. (A.3)). The discrepancy in the sizes is much more pronounced for higher memory.

Taking the inverse of the smallest root of the denominator we have

$$\mu^{(d)} \leqslant \mu_4^{(d)} = \frac{6R}{R^2 - 56d + 40 + 16d^2 - 4dR + 4R}$$

where

$$R = \sqrt[3]{336d^2 - 480d + 316 - 64d^3 + 12\sqrt{192d^3 - 72d^2 - 48d^4 - 240d + 249}}$$

7. MEMORY = 6

There are 9 mistake classes for memory = 6. Their canonical mistakes are $[1, -1]$, $[1, 2, -1, -2]$, $[1, 2, -1, -1, -2, 1]$, $[1, 1, 2, -1, -1, -2]$, $[1, 2, 2, -1, -2, -2]$, $[1, 2, 3, -1, -3, -2]$, $[1, 2, 3, -2, -1, -3]$, $[1, 2, -1, 3, -2, -3]$ and $[1, 2, 3, -1, -2, -3]$. Some of these mistakes do not exist in dimension 2. However, we are using these to canonically represent specific mistake classes which happen to be empty in dimension 2. For example, in dimension 2, the mistake class represented by the mistake $[1, 2, 3, -1, -2, -3]$ is empty. So we proceed unhindered.

Definition 7.1. For a walk, w, let $\dim(w)$ denote the number of dimensions that w spans.

Thus $\dim([1, 2, 3, -2, -3, 1, 3]) = 3$, $\dim([9, 8, -9, -9]) = 2$ and $\dim([2, 2, 2, 2]) = 1$.

Definition 7.2. Let

$$Y_{\mathfrak{p}} := \sum_{\mathfrak{m} \mid \mathfrak{p} \text{ is a suffix of } \mathfrak{m}} 2^{\dim(\mathfrak{m}) - \dim(\mathfrak{p})} \binom{d - \dim(\mathfrak{p})}{\dim(\mathfrak{m}) - \dim(\mathfrak{p})} \times (\dim(\mathfrak{m}) - \dim \mathfrak{p}))!\, C_{[\mathfrak{m}]}$$

There are exactly $2^{\dim(\mathfrak{m}) - \dim(\mathfrak{p})}\binom{d - \dim(\mathfrak{p})}{\dim(\mathfrak{m}) - \dim(\mathfrak{p})}(\dim(\mathfrak{m}) - \dim(\mathfrak{p}))!$ mistakes similar to $\mathfrak{m}$ which have the suffix $\mathfrak{p}$ exactly. This is due to the fact that every mistake similar to $\mathfrak{m}$ is the result of the action of a signed permutation on $\mathfrak{m}$. Every mistake that has the suffix $\mathfrak{p}$ exactly must be the result of mapping the last $\gamma(\mathfrak{p})$ steps of $\mathfrak{m}$ to those of $\mathfrak{p}$. There are $\dim(\mathfrak{m}) - \dim(\mathfrak{p})$ directions that have not been mapped and these may be mapped to any direction other than those specified in $\mathfrak{p}$. Thus there are $2^{\dim(\mathfrak{m}) - \dim(\mathfrak{p})}\binom{d - \dim(\mathfrak{p})}{\dim(\mathfrak{m}) - \dim(\mathfrak{p})}(\dim(\mathfrak{m}) - \dim(\mathfrak{p}))!$ different mappings each of which produces a different mistake similar to $\mathfrak{m}$ which has the suffix $\mathfrak{p}$ exactly.

We write

$$
\begin{aligned}
C_{[1,-1]} &= -s^2 - sY_{[1]} \\
C_{[1,2,-1,-2]} &= -s^4 - s^3 Y_{[1]} - s^2 Y_{[1,2]} - sY_{[1,2,-1]} \\
C_{[1,1,2,-1,-1,-2]} &= -s^6 - s^5 Y_{[1]} - s^4 Y_{[1,1]} - s^3 Y_{[1,1,2]} \\
&\quad - s^2 Y_{[1,1,2,-1]} - sY_{[1,1,2,-1,-1]} \\
C_{[1,2,2,-1,-2,-2]} &= -s^6 - s^5 Y_{[1]} - s^4 Y_{[1,2]} - s^3 Y_{[1,2,2]} \\
&\quad - s^2 Y_{[1,2,2,-1]} - sY_{[1,2,2,-1,-2]} \\
C_{[1,2,-1,-1,-2,1]} &= -s^6 - s^5 Y_{[1]} - s^4 Y_{[1,2]} - s^3 Y_{[1,2,-1]} \\
&\quad - s^2 Y_{[1,2,-1,-1]} - sY_{[1,2,-1,-1,-2]} \\
C_{[1,2,3,-1,-2,-3]} &= -s^6 - s^5 Y_{[1]} - s^4 Y_{[1,2]} - s^3 Y_{[1,2,3]} \\
&\quad - s^2 Y_{[1,2,3,-1]} - sY_{[1,2,3,-1,-2]} \\
C_{[1,2,-1,3,-2,-3]} &= -s^6 - s^5 Y_{[1]} - s^4 Y_{[1,2]} - s^3 Y_{[1,2,-1]} \\
&\quad - s^2 Y_{[1,2,-1,3]} - sY_{[1,2,-1,3,-2]} \\
C_{[1,2,3,-2,-1,-3]} &= -s^6 - s^5 Y_{[1]} - s^4 Y_{[1,2]} - s^3 Y_{[1,2,3]} \\
&\quad - s^2 Y_{[1,2,3,-2]} - sY_{[1,2,3,-2,-1]} \\
C_{[1,2,3,-1,-3,-2]} &= -s^6 - s^5 Y_{[1]} - s^4 Y_{[1,2]} - s^3 Y_{[1,2,3]} \\
&\quad - s^2 Y_{[1,2,3,-1]} - sY_{[1,2,3,-1,-3]}
\end{aligned}
$$

Here, the Y's are defined as above, for example,

$$
\begin{aligned}
Y_{[1]} = {} & C_{[1,-1]} + 2(d-1)\, C_{[1,2,-1,-2]} + 2(d-1)\, C_{[1,1,2,-1,-1,-2]} \\
& + 2(d-1)\, C_{[1,2,2,-1,-2,-2]} \\
& + 2(d-1)\, C_{[1,2,-1,-1,-2,1]} + 4(d-1)(d-2)\, C_{[1,2,3,-1,-2,-3]} \\
& + 4(d-1)(d-2)\, C_{[1,2,-1,3,-2,-3]} \\
& + 4(d-1)(d-2)\, C_{[1,2,3,-2,-1,-3]} \\
& + 4(d-1)(d-2)\, C_{[1,2,3,-1,-3,-2]}
\end{aligned}
$$

When we solve this system, and make the necessary substitutions in the generating function, we find that

$$
\begin{aligned}
F_6^{(d)} = &-((4*\mathrm{d}\wedge 2-4*\mathrm{d})*\mathrm{s}\wedge 19\\
&+(-16*\mathrm{d}\wedge 2+10*\mathrm{d}+8*\mathrm{d}\wedge 3-1)*\mathrm{s}\wedge 18\\
&+(-48*\mathrm{d}\wedge 2+16*\mathrm{d}\wedge 3+34*\mathrm{d}+1)*\mathrm{s}\wedge 17\\
&+(-8*\mathrm{d}\wedge 3+10+24*\mathrm{d}\wedge 2-26*\mathrm{d})*\mathrm{s}\wedge 16\\
&+(92*\mathrm{d}\wedge 2-64*\mathrm{d}-3-32*\mathrm{d}\wedge 3)*\mathrm{s}\wedge 15\\
&+(48*\mathrm{d}\wedge 2-24*\mathrm{d}\wedge 3-28-2*\mathrm{d})*\mathrm{s}\wedge 14\\
&+(-8*\mathrm{d}\wedge 3+4*\mathrm{d}\wedge 2+20*\mathrm{d}-14)*\mathrm{s}\wedge 13\\
&+(17-8*\mathrm{d}\wedge 2)*\mathrm{s}\wedge 12\\
&+(-12*\mathrm{d}\wedge 2+8*\mathrm{d}\wedge 3+48-36*\mathrm{d})*\mathrm{s}\wedge 11\\
&+(-44*\mathrm{d}\wedge 2+16*\mathrm{d}\wedge 3+14*\mathrm{d}+16)*\mathrm{s}\wedge 10\\
&+(-22+32*\mathrm{d}\wedge 3+102*\mathrm{d}-108*\mathrm{d}\wedge 2)*\mathrm{s}\wedge 9\\
&+(50*\mathrm{d}-64*\mathrm{d}\wedge 2-9+24*\mathrm{d}\wedge 3)*\mathrm{s}\wedge 8\\
&+(-60*\mathrm{d}+36*\mathrm{d}\wedge 2+7)*\mathrm{s}\wedge 7\\
&+(-24*\mathrm{d}-26+28*\mathrm{d}\wedge 2)*\mathrm{s}\wedge 6\\
&+(-19+16*\mathrm{d}\wedge 2-4*\mathrm{d})*\mathrm{s}\wedge 5\\
&+(-9+18*\mathrm{d})*\mathrm{s}\wedge 4+(4*\mathrm{d}+10)*\mathrm{s}\wedge 3+9*\mathrm{s}\wedge 2+4*\mathrm{s}+1)\\
&/(\mathrm{s}\wedge 18+(2*\mathrm{d}-1)*\mathrm{s}\wedge 17+(-10+6*\mathrm{d})*\mathrm{s}\wedge 16\\
&+(3+4*\mathrm{d}\wedge 2-10*\mathrm{d})*\mathrm{s}\wedge 15\\
&+(28+8*\mathrm{d}\wedge 2-32*\mathrm{d})*\mathrm{s}\wedge 14+(-10*\mathrm{d}+14)*\mathrm{s}\wedge 13\\
&+(32*\mathrm{d}-17-12*\mathrm{d}\wedge 2)*\mathrm{s}\wedge 12\\
&+(-48-20*\mathrm{d}\wedge 2+64*\mathrm{d})*\mathrm{s}\wedge 11\\
&+(22*\mathrm{d}-16-8*\mathrm{d}\wedge 2)*\mathrm{s}\wedge 10\\
&+(22-22*\mathrm{d}+4*\mathrm{d}\wedge 2)*\mathrm{s}\wedge 9\\
&+(9+8*\mathrm{d}\wedge 2-24*\mathrm{d})*\mathrm{s}\wedge 8+(-12*\mathrm{d}-7+8*\mathrm{d}\wedge 2)*\mathrm{s}\wedge 7\\
&+(26+12*\mathrm{d}\wedge 2-32*\mathrm{d})*\mathrm{s}\wedge 6\\
&+(19+8*\mathrm{d}\wedge 2-18*\mathrm{d})*\mathrm{s}\wedge 5+(4*\mathrm{d}\wedge 2-6*\mathrm{d}+9)*\mathrm{s}\wedge 4\\
&+(8*\mathrm{d}-10)*\mathrm{s}\wedge 3+(-9+6*\mathrm{d})*\mathrm{s}\wedge 2+(2*\mathrm{d}-4)*\mathrm{s}-1)
\end{aligned}
$$

8. TAYLOR EXPANSIONS

Computing the *exact* generating function for a given memory and general dimension can be cumbersome, we can simplify the task for greater memories by specifiying the dimension we are interested in, for example;

$$
\begin{aligned}
F_8^{(2)} = &-(8*s\wedge 46+24*s\wedge 45+32*s\wedge 44+24*s\wedge 43+11*s\wedge 42\\
&-33*s\wedge 41-80*s\wedge 40-72*s\wedge 39\\
&-16*s\wedge 38+52*s\wedge 37+101*s\wedge 36+141*s\wedge 35+49*s\wedge 34\\
&-54*s\wedge 33-117*s\wedge 32-94*s\wedge 31\\
&-117*s\wedge 30+61*s\wedge 29+47*s\wedge 28+16*s\wedge 27+2*s\wedge 26\\
&+24*s\wedge 25-82*s\wedge 24+50*s\wedge 23\\
&+141*s\wedge 22+74*s\wedge 21+84*s\wedge 20-4*s\wedge 19-61*s\wedge 18\\
&-153*s\wedge 17-117*s\wedge 16-132*s\wedge 15\\
&-36*s\wedge 14-s\wedge 13-21*s\wedge 12+22*s\wedge 11\\
&+5*s\wedge 10+28*s\wedge 9\\
&-18*s\wedge 8+41*s\wedge 7+9*s\wedge 6\\
&+14*s\wedge 5+6*s\wedge 4+6*s\wedge 3+3*s\wedge 2+2*s+1)\\
&/(s\wedge 42+s\wedge 41-4*s\wedge 37-s\wedge 36+3*s\wedge 35+3*s\wedge 34\\
&+2*s\wedge 33+5*s\wedge 32-2*s\wedge 31-7*s\wedge 30\\
&-5*s\wedge 29-3*s\wedge 28+4*s\wedge 27+2*s\wedge 26+4*s\wedge 25\\
&-2*s\wedge 24+2*s\wedge 23-13*s\wedge 22+6*s\wedge 21\\
&+4*s\wedge 20+12*s\wedge 19+s\wedge 18+9*s\wedge 17-7*s\wedge 16\\
&-8*s\wedge 15-8*s\wedge 14-7*s\wedge 13+s\wedge 12\\
&-2*s\wedge 11+3*s\wedge 10-4*s\wedge 9+2*s\wedge 8-5*s\wedge 7-s\wedge 6\\
&+2*s\wedge 5+2*s\wedge 4+2*s\wedge 3+s\wedge 2+2*s-1)
\end{aligned}
$$

This, however, also becomes quite complicated, as the above function demonstrates.

Instead of calculating the exact generating function for dimension d and memory k, we will merely compute a large number of terms of the Taylor expansion of the generating function. This will allow us to do two things.

First, we will have an exact enumeration for the number of walks on dimension d that are self-avoiding with memory k up to a certain number of steps (however far we expand the generating function). More important is that our Taylor expansion will allow us to compute a bound for the connective constant for walks on dimension d with memory k. Hence, we will have a bound for self-avoiding walks on dimension d.

We proceed by extracting the necessary information systematically from the cluster generating functions, $C_{[\mathfrak{m}]}$.

From (2), (3) and (4) we have

$$F_k^{(d)} = \frac{1}{1 - 2ds - \sum_{\mathfrak{m} \in M_k} \binom{d}{\dim(\mathfrak{m})} (\dim(\mathfrak{m}))!\, 2^{\dim(\mathfrak{m})} C_{[\mathfrak{m}]}}$$

If we want to know the coefficient of s^i of $F_k^{(d)}$, we need only determine the coefficients of $C_{[\mathfrak{m}]}$ up to s^i. If we carefully examine the equations for the cluster generating functions, we see that to compute the coefficient of s^i in $C_{[\mathfrak{m}]}$, we only need to keep track of the coefficients of s^j of the other cluster generating functions for $i - k \leqslant j < i$. In fact, we need not keep track of $C_{[\mathfrak{m}]}$ at all if we have sufficient information about the $Y_{\mathfrak{p}}$'s. It is this strategy that we adopt to compute the coefficient of $F_k^{(d)}$ for there is much less overhead in the computation of the Y's than that of the C's. In order to compute the C's, we need information about the interaction of every C with every other C. In order to compute the Y's, we need only keep a list of mistakes somewhere in memory. $C_{[\mathfrak{m}]}$ contributes to $Y_{\mathfrak{p}}$ only if $\mathfrak{p}$ is a suffix of $\mathfrak{m}$. Likewise, $Y_{\mathfrak{p}}$ contributes to $C_{[\mathfrak{m}]}$ only if $\mathfrak{p}$ is a prefix of $\mathfrak{m}$. The result of this manipulation allows us to compute the expansion of the generating function quickly and through clever programming, we can eliminate the need to compute the C's altogether, speeding our computation even more.

The resulting set of equations are of the form

$$Y_{\mathfrak{p}} = \sum_{\mathfrak{m} \in M_{k'},\, k' \leqslant k \mid \mathfrak{p} \text{ is equivalent to a suffix of } \mathfrak{m}} 2^{\dim(\mathfrak{m}) - \dim(\mathfrak{p})} \binom{d - \dim(\mathfrak{p})}{\dim(\mathfrak{m}) - \dim(\mathfrak{p})} \times (\dim(\mathfrak{m}) - \dim(\mathfrak{p}))!\, C_{[\mathfrak{m}]}$$

for all mistake suffices, $\mathfrak{p}$ and

$$C_{[\mathfrak{m}]} = \sum_{\mathfrak{p} \mid \mathfrak{p} \text{ is equivalent to a prefix of } \mathfrak{m}} (-1)\, s^{\lambda(\mathfrak{p})}\, Y_{\mathfrak{p}}$$

for all $\mathfrak{m} \in M_{k'}$, $k' \leqslant k$.

The number of mistakes we are allowing for is finite, hence the generating function $F_k^{(d)}$ must be rational. Furthermore, by the Perron–Frobenius theorem, the smallest root of the denominator of that rational function must be simple. As a result, the coefficients of the Taylor expansion of the $F_k^{(d)}$ are asymptotic to $C\alpha^N$ where α is the smallest positive root of the denominator of $F_k^{(d)}$. The ratios of consecutive coefficients converges to α which in our case is the connective constant for walks with finite memory.

9. FURTHER RESULTS

It is possible to apply this method to other lattices by adding strategic mistakes. For example, the hexagonal lattice is nothing more than the simple cubic lattice on 3 dimensions with certain bonds missing. If we define these bonds as "mistakes, we can compute the generating function for walks on this lattice. Here, we will take the following polygon in dimension 3 to be the basis for the hexagonal lattice: 1, 2, 3, −1, −2, −3. Indeed, the two dimensional representation of this polygon is a hexagon. Thus the mistakes are [1, 1], [1, −2], [1, 3], [2, −1], [2, 2], [2, −3], [3, 1], [3, −2], [3, 3], [−1, −1], [−1, 2], [−1, −3], [−2, 1], [−2, −2], [−2, 3], [−3, −1], [−3, 2], [−3, −3]. Note that if we begin at the origin and disallow these steps, we actually end up with 2 distinct sets of walks, one using the bonds mentioned above, and the other on the compliment of those bonds. This is because of the fact that our method cannot impose a condition on the first step of the walk, which in this case would have to be either 1, 3, or −2. So we must divide each term of our generating function by 2 except the s^0 term. We achieve this by dividing by 2 then adding $\frac{1}{2}$. The author has as yet achieved no new bounds for this lattice. It is hoped that this method may be applied to this (and other) lattices to achieve further results.

10. SUMMARY OF RESULTS

Table 1 shows the best results we have obtained thus far for upper bounds of the connective constant in dimensions 2–6. The memories are given as well as the term used(n) for obtaining the ration given.

The numbers under the "nth root heading in the above table were computed using the bound $(c_n)/(c_1)^{1/(n-1)}$ which is attributed to Alm in (15). Here we use $c_{n,k}$. As was stated earlier, $c_{n,k} > c_n$ and so the bound still holds.

The numbers under the "ratios" heading in the above table are the limits of $\mu_k^{(d)} = \lim_{n\to\infty}(c_{n,k}^{(d)})/(c_{n-1,k}^{(d)})$, where $c_{n,k}^d$ is the number of finite-memory, with memory k, n-step self-avoiding walks in Z^d. By the Perron–Frobenius theorem, we know (since for finite memory we have a finite

Table 1. Computed Bounds for the Connective Constants

			New bounds			
Dimension	Estimate	Previous bound	Ratios	nth root	Memory	n
2	2.6381585[3]	2.6958[1]	2.6939	2.7054	16	158
3	4.6839066[11]	4.7560[1]	4.7476	4.7520	14	108
4	6.7720[7]	6.8320[1]	6.8179	6.8188	10	648
5	8.83861[8]	8.8808[1]	8.8602	8.8608	10	698
6	10.87879[8]	10.9025[1]	10.8886	10.8893	10	448

Markov Chain with a finite transfer matrix, with non-negative coefficients), that $c_{n,k}^{(d)} = C(\mu_k^{(d)})^n + O(\nu^n)$, for some constant C (that depends of course on d and k, but not on n, and $|\nu| \leqslant \mu$), and hence the above ratios converge exponentially fast to $\mu_k^{(d)}$. This is corroborated by the computer output. For example the ratios for $d=2$, and $k=16$, all the ratios for $n=151,\ldots,158$ are equal to 2.693848921595418498205362276245 1.... This is in sharp contrast to the situation for genuine self-avoiding walks, where the asymptotics is known, and the conjectured asymptotics imply very slow convergence of the ratios.

While we do not have rigorous error bounds for the rate of convergence, it is clear that the series expansion, for any fixed memory, $k=16$ in our case, could be continued indefinitely with a polynomial time (in fact $O(1)$ per term, and counting bit-operations, $O(\log n)$), per each extra term, so eventually we would be able to use the nth root bound to any desired precision, with a polynomial cost. (The analogous situation for SAWs requires exponential effort!)

However, with a larger computer, and a more efficient implementation in a lower-level programming language, such as C, one would be able, with the present approach, to achieve higher memories, that would improve the present upper bounds even further, we did not bother to go that far. If one wishes to go really far, one can use modular methods and the Chinese Remainder Theorem, as it was done in ref. 2, which also has the advantage that the computations are parallelizable.

Comparing our results with previous results, we see the largest gains are in high dimensions. It is hoped that extensions of this method may yield better gains in dimensions 2 and 3. One such extension would be the construction of a set of mistakes (in dimension 2 or 3) for which we are able to easily determine the interactions which are necessary for implementing this method. In dimension 2, the set of all rectangles up to a given size, say up to $n \times n$, is one set of mistakes which allow an easy implementation of this

method. The bound obtained from the generating function for walks avoiding such a list of mistakes is too large to be of interest in this context.

Other results obtained from this study include the exact enumeration of walks that are self-avoiding with certain memories. Tables of these results are too massive for printing but may be obtained at the World Wide Web site listed below.

Note. A small Maple package accompanying this paper, saw.maple, can be obtained by using your favorite world wide web browser at http://www.math.temple.edu/~noonan/saw/ or by anonymous ftp to ftp.math.temple.edu, directory /pub/noonan. Exact enumerations, generating functions and other information may also be obtained at the above site.

ACKNOWLEDGMENTS

I would like to thank Sven Alm, Tony Guttmann, Neal Madras, and Gordon Slade, and one of the referees for very helpful comments and for graciously sharing their work with me. This work is part of the author's Ph.D. thesis, conducted under the direction of Doron Zeilberger at Temple University.

REFERENCES

1. S. E. Alm, Upper bounds for the connective constant of self-avoiding walks, *Combinatorics, Probability and Computing* **2**:115–136 (1993).
2. A. R. Conway and A. J. Guttmann, Square lattice self-avoiding walks and corrections-to-scaling, *Physics Review Letters* (to appear).
3. I. G. Enting and A. J. Guttmann, The size and number of rings on the square lattice, *J. Phys. A* **21**:L165–L172 (1988).
4. M. Fisher and D. Gaunt, Ising model and self-avoiding walks on the hypercubical lattices and "high-density" expansions, *Physical Review A* **113**:224–239 (1984).
5. M. Fisher and Sykes, Excluded-volume problem and the Ising model of ferromagnetism, *Phys. Rev.* **114**:45–58 (1959).
6. I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration* (Wiley, New York, 1983).
7. A. J. Guttmann, On the zero field susceptibility in the $d=4$, $n=0$ limit; analyzing for confluent logarithmic singularities, *J. Phys. A* **11**:L103–L106 (1978).
8. A. J. Guttmann, Correction to scaling exponents and critical properties of the n-vector model with dimensionality >4, *J. Phys. A* **14**:233–239 (1981).
9. A. J. Guttmann, Bounds on connective constants for self-avoiding walks, *J. Phys. A* **16**:2233–2238 (1983).
10. A. J. Guttmann, Bounds on self-avoiding walks on directed square lattices, *J. Phys. A* **16**:3885–3984 (1983).
11. A. J. Guttmann, On the critical behavior of self-avoiding walks, *J. Phys. A* **20**:1839–1854 (1987).

12. J. M. Hammersley, Percolation processes II. The connective constant, *Proc. Camb. Phil. Soc.* **53**:643–645 (1957).
13. T. Hara and G. Slade, The lace expansion for self-avoiding walk in five or more dimensions, *Reviews in Math. Phys.* **4**:101–136 (1992).
14. W. MacPhail, A walk into the fifth dimension (to gain a better understanding of polymers), *The Hamilton Spectator* (Jan. 26, 1991).
15. Madras and Slade, *The Self-Avoiding Walk* (Birkhäuser, Boston, 1993).
16. J. Noonan and D. Zeilberger, The Goulden-Jackson method: Extensions, Applications and Implementations (in preparation).
17. G. Slade, Gordon, Bounds on the Self-Avoiding Walk Connective Constant, *Jour. Four. Anal. Appl.* 525–533 (1995).
18. G. Slade, Random Walks, *American Scientist* **84**:146–153 (March 1996).