The dimensions of knotted polygons

1991 J. Phys. A: Math. Gen. 24 3935

(http://iopscience.iop.org/0305-4470/24/16/028)

J. Phys. A: Math. Gen. **24** (1991) 3935-3948. Printed in the UK

The dimensions of knotted polygons

E J Janse van Rensburg† and S G Whittington‡

† Supercomputer Computations Research Institute, Florida State University, Tallahassee, FL 32306-4052, USA

‡ Department of Chemistry, University of Toronto, Toronto, Ontario, Canada M5S 1A1

Received 25 February 1991, in final form 8 April 1991

Abstract. We study the dimensions (mean-square radius of gyration and mean span) of self-avoiding polygons on the simple cubic lattice with fixed knot type. The approach used is a Monte Carlo algorithm which is a combination of the BFACF algorithm and the pivot algorithm, so that the polygons are studied in the grand canonical ensemble, but the autocorrelation time is not too large. We show that, although the dimensions of polygons are sensitive to knot type, the critical exponent (ν) and the leading amplitude are independent of the knot type of the polygon. The knot type influences the confluent correction to scaling term and hence the rate of approach to the limiting behaviour.

1. Introduction

Linear polymers in dilute solution can be highly self-entangled and these entanglements can influence crystallization behaviour (de Gennes 1984). If a ring closure reaction occurs the entanglement can be trapped as a knot in the resulting ring polymer, and some information about the entanglement complexity can be obtained from a study of the distribution of knots in the ring polymer. In addition, the presence of knots in closed circular DNA can given information about the mechanism of action of enzymes acting on the DNA molecule (Wasserman *et al* 1985, Wasserman and Cozzarelli 1986, Sumners 1987, 1990).

Ring polymers can be modelled as self-avoiding polygons on a (three-dimensional) lattice, and the presence of knots in these (and related) models has been studied using Monte Carlo techniques by a number of workers (Vologodskii *et al* 1974, Frank-Kamenetskii *et al* 1975, Michels and Wiegel 1984, 1986, Janse van Rensburg and Whittington 1990). Rather little is known rigorously, but it has been shown that sufficiently long polygons are knotted with probability one (Sumners and Whittingon 1988, Pippenger 1989). The influence of knots on the dimensions of polygons has not received much attention, though this seems important in understanding the separation of DNA molecules with different knot types by pulsed gel electrophoresis techniques.

We examine this problem here using a Metropolis (Metropolis *et al* 1953) style Monte Carlo approach involving sampling on a realization of a Markov chain. We use an algorithm which is a combination of the BFACF and pivot algorithms. The BFACF algorithm consists of local moves which can change the length of the polygon but not the knot type. (In fact it can be shown that the ergodic classes are just the knot types, so that the algorithm samples polygons with fixed knot type and every such polygon can be obtained in the sample.) However, the algorithm has the disadvantage that it has very long autocorrelation times, which makes it unsuitable for a Monte Carlo study

0305-4470/91/163935+14$03.50

of very long polygons. To remedy this situation, we use an algorithm which is a combination of BFACF *local* moves and large scale pivot moves. This introduces a new problem since a pivot move can change the knot type of the polygon. After each successful pivot move we compute the Alexander polynomial (Frank-Kamenetskii *et al* 1975, Michels and Wiegel 1984, 1986, Burde and Zieschang 1985) of the polygon, and reject the move if the Alexander polynomial changes. (Of course, this is not a complete solution to the difficulty since distinct knots can have the same Alexander polynomial. For instance the knot 10_{132} has the same Alexander polynomial as the knot 5_1, and the knot 8_{20} has the same Alexander polynomial as the knot $3_1 3_1$. However, the computation of other knot invariants would be prohibitively time consuming and we do not believe that significant errors are introduced by this approximation.)

Using this algorithm we generate samples of polygons with some simple knot types, and estimate their mean-square radii of gyration and mean span as a function of the number of edges in the polygon. From these results we estimate the corresponding critical exponents and amplitudes.

2. The BFACF algorithm and knotted polygons

An unrooted self-avoiding polygon, or *polygon*, in any lattice, is a sequence of lattice sites $\omega_0, \omega_1, \omega_2, \ldots, \omega_n$, and associated edges (ω_i, ω_{i+1}) such that $\omega_0 = \omega_n$, and ω_i and ω_{i+1} are nearest neighbours in the lattice, and $\omega_1, \omega_2, \ldots, \omega_n$ are all distinct. If ω_0 and ω_n are not the same, then we have a *self-avoiding walk*. Let $\mathscr{Z}^d$ be the d-dimensional hypercubic lattice and let $\{e_i\}_{i=1}^d$ be the set of orthogonal unit vectors in $\mathscr{Z}^d$. The BFACF algorithm is a local stochastic process which operates on paths (any sequence of edges) in the hypercubic lattice. It generates statistical ensembles of paths with a Boltzmann distribution (Berg and Foester 1981). The algorithm was first applied to 'bosonic' walks (Brownian walks) and 'fermionic' walks (Brownian walks without 'spikes') by Berg and Foester (1981), before it was applied to self-avoiding walks by Aragao de Carvalho *et al* (1983) and Aragao de Carvalho and Caracciolo (1983).

Let P be the set of all polygons in $\mathscr{Z}^d$. Let $\omega \in P$. Then the BFACF algorithm is defined in the following manner: pick an edge (ω_i, ω_{i+1}) of the current polygon with uniform probability. Pick a unit vector e_i perpendicular to (ω_i, ω_{i+1}). Move the chosen edge one lattice space along e_i, inserting two new edges at its endpoints to keep the polygon intact. Lastly, erase any double edges (spikes) which may result from this process. A little reflection indicates that these operations result in one of the three moves illustrated in figure 1, with possible changes in the length of the polygon of ± 2 or 0. Let the new configuration be ν. Accept ν with probability $p(\omega \to \nu) = \chi(\nu) Q(\beta^2)$, where χ is an indicator which is 1 if ν is a polygon, and 0 otherwise, and $Q(\beta^2) = 1$ if the length change is 0 or -2, and otherwise $Q(\beta^2) = \beta^2$.

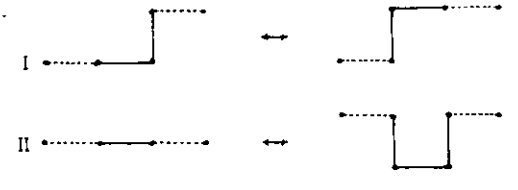

Figure 1. The two elementary BFACF moves.

It is easy to check that this Metropolis implementation (Metropolis *et al* 1953) of the BFACF algorithm is reversible. In three dimensions the ergodicity classes of the algorithm are the knot-types of the polygons (Janse van Rensburg and Whittington 1991). The BFACF algorithm for unrooted polygons realizes a Markov chain with the invariant probability distribution

$$\pi_\beta(\omega) = Z(\beta)^{-1}|\omega|\beta^{|\omega|} \tag{2.1}$$

where $\omega \in \mathcal{K}$, the set of all polygons of a given knot-type. $|\omega|$ is the number of edges in ω and β is an adjustable parameter. The partition function of the ensemble of polygons generated by the algorithm is

$$Z(\beta) = \sum_{\omega \in \mathcal{K}} |\omega|\beta^{|\omega|} \tag{2.2}$$

where the set $\mathcal{K}$ is determined by the initial polygon in the Markov chain.

The basic elementary transition is described by a transition probability matrix $\mathbf{P} = \{p(\omega \to \nu)\} = \{p_{\omega\nu}\}$ which has the following properties in its ergodic classes: (i) For each ω, $\nu \in \mathcal{K}$ there exists an $m \geq 0$ such that the m-step transition probability from ω to ν is positive. (This is obviously true in any given ergodic class.) (ii) For each polygon $\omega \in \mathcal{K}$, $\Sigma_{\omega \in \mathcal{K}} \pi_\omega p_{\omega\nu} = \pi_\nu$. We can easily check this with our choice of transition probabilities. Therefore, it can be shown that π_ω is the unique limit distribution of the Markov chain with state space $\mathcal{K}$ and transition probability matrix $\mathbf{P}$ (Kemeny and Snell 1976). Let the observed states of a realization of this Markov chain be represented by X_i. Then the sequence of states $\{X_i\}$ is in general correlated. An observable $A_t = A(X_t)$ is a stationary stochastic process with mean

$$\langle A_t \rangle = \sum_{\omega \in \mathcal{K}} \pi_\omega A(\omega) \tag{2.3}$$

and unnormalized autocorrelation function

$$C_{AA}(s) = \langle A_t A_{t+s} \rangle - \langle A_t \rangle^2. \tag{2.4}$$

The normalized autocorrelation is defined by $\rho_{AA}(t) = C_{AA}(t)/C_{AA}(0)$. Once the Markov chain is in equilibrium, we define the *integrated autocorrelation time* τ_{int} by

$$\tau_{\text{int}}(A) = \tfrac{1}{2} \sum_{t=-\infty}^{\infty} \rho_{AA}(t). \tag{2.5}$$

The variance in the sample mean of the observable A_t over N observations is asymptotically

$$\sigma^2(A) \sim \frac{2\tau_{\text{int}}(A)}{N} C_{AA}(0). \tag{2.6}$$

In other words, the effective number of 'independent' observations is $N/2\tau_{\text{int}}(A)$. The integrated autocorrelation time controls the statistical error in our measurement of the sample mean of A_t.

In general, the calculation of the integrated autocorrelation time τ_{int} over a finite set of data can be performed in the manner suggested by Madras and Sokal (1988). A window $\lambda(t)$, which has $\lambda(t) = 1$ if $|t| < T$ and zero otherwise, is chosen, and equation (2.5) is approximated by

$$\bar{\tau}_{\text{int}} = \tfrac{1}{2} \sum_{t=-N}^{N} \rho_{AA}(t)\lambda(t). \tag{2.7}$$

In this equation we exclude contributions to the autocorrelation time which consist mostly of noise. We have a sequence of $2N+1$ measurements and, for this approximation to be good, we must have $0 \ll T \ll N$. For details, see Madras and Sokal (1988).

3. Implementation

3.1. A biased version of the BFACF *algorithm*

In this paper we are mainly interested in the size of polygons of a fixed knot type. This can be investigated by studying the mean square radius of gyration and the mean span of a representative sample of polygons. We define the square radius of gyration of a polygon ω by

$$r^2(\omega) = \frac{1}{n} \sum_{i=0}^{n-1} ((X(\omega_i) - \bar{X}(\omega))^2 + (Y(\omega_i) - \bar{Y}(\omega))^2 + (Z(\omega_i) - \bar{Z}(\omega))^2) \tag{3.1}$$

where $X(x)$, $Y(x)$ and $Z(x)$ are the first, second and third components of the vertex x, and $\bar{X}(\omega)$, $\bar{Y}(\omega)$ and $\bar{Z}(\omega)$ are the components of the centre of mass of the polygon ω. The span of a polygon is defined by

$$s(\omega) = \tfrac{1}{3}(\max_{ij} |X(\omega_i) - X(\omega_j)| + \max_{ij} |Y(\omega_i) - Y(\omega_j)| + \max_{ij} |Z(\omega_i) - Z(\omega_j)|). \tag{3.2}$$

The mean square radius of gyration and the mean span are measured in terms of a single length scale in the problem. We therefore assume that for polygons of knot-type $\mathcal{K}$,

$$\langle r^2 \rangle \approx C_{\mathcal{K}} n^{2\nu_{\mathcal{K}}} \tag{3.3}$$

and

$$\langle s \rangle \approx D_{\mathcal{K}} n^{\nu_{\mathcal{K}}}. \tag{3.4}$$

The asymptotic behaviour can therefore be characterized by studying the quantities $\nu_{\mathcal{K}}$ and $C_{\mathcal{K}}$ and $D_{\mathcal{K}}$.

This task requires the sampling of polygons of sufficient length, so that the scaling of the quantities in equations (3.3) and (3.4) can be studied effectively. The partition function of polygons generated by the BFACF algorithm as described in the previous section can be written as

$$Z(\beta) = \sum_{n=0}^{\infty} n p_n(\mathcal{K}) \beta^n \tag{3.5}$$

where $p_n(\mathcal{K})$ is the number of polygons of the knot-type $\mathcal{K}$ and with n edges. It is now well established that

$$\sum_{\mathcal{K}} p_n(\mathcal{K}) \approx n^{\alpha - 3} \mu^n \tag{3.6}$$

where μ is a growth-constant and α is an exponent with value close to 0.2 (Le Guillou and Zinn-Justin 1980, 1989), if the polygons are not rooted. If we substitute equation (3.6) into (2.1), then we see that the BFACF algorithm, when applied to polygons, will produce small polygons most of the time. This is a drawback for the application we have in mind in this paper and, in our implementation, we bias the algorithm towards

longer polygons. The simplest way of doing this is to make the parameter β dependent on n. We choose the following dependence:

$$\beta = \begin{cases} \beta_0 & \text{if } n \leqslant n_{\min} \\ \beta_1 & \text{if } n_{\min} < n < n_{\max} \\ \beta_2 & \text{if } n \geqslant n_{\max}. \end{cases} \tag{3.7}$$

We can choose the values of the numbers β_i such that most of the polygons generated by the algorithm are between the limits $n_{\min}$ and $n_{\max}$. In our applications we chose $\beta_1 = 0.213\,475$, which is close to the critical value which is believed to be 0.213 496 (Guttmann 1989). The value of β_0 was $1.05\beta_1$ and we took β_2 to be equal to $0.95\beta_1$. This choice for β preserves reversibility and, while some regions of phase space are now less likely to be visited, most of the computer time is spent in regions of phase space which are of interest to us. The Markov chain is still ergodic and has the limit distribution (if we assume that $p_n(\mathscr{K}) \approx n^{\alpha_{\mathscr{K}}-3}\mu_{\mathscr{K}}^n$)

$$\pi_\beta(\omega) = Z(\beta)^{-1}|\omega|^{\alpha_{\mathscr{K}}-2} \times \begin{cases} (\beta_0\mu_{\mathscr{K}})^{|\omega|} & \text{if } |\omega| \leqslant n_{\min} \\ (\beta_0\mu_{\mathscr{K}})^{n_{\min}}(\beta_1\mu_{\mathscr{K}})^{|\omega|-n_{\min}} & \text{if } n_{\min} < |\omega| < n_{\max} \\ (\beta_0\mu_{\mathscr{K}})^{n_{\min}}(\beta_1\mu_{\mathscr{K}})^{n_{\max}-n_{\min}}(\beta_2\mu_{\mathscr{K}})^{|\omega|-n_{\max}} & \text{if } |\omega| \geqslant n_{\max}. \end{cases} \tag{3.8}$$

The partition function $Z(\beta)$ is given by $Z(\beta) = \Sigma_\omega \pi_\beta(\omega)$. We observe that polygons with the same lengths have the same weight in equation (3.8). For every set of polygons of fixed length, the algorithm will generate a representative sample with a uniform distribution. We can therefore use this biased algorithm to study equations (3.3) and (3.4).

3.2. Numerical details

The algorithm was programmed in C and the numerical work performed on an Apollo DN10000. The most natural storage of a discrete object in computer memory is as an unordered list of vertices. The ith address in the list contains in its first three elements the coordinates of the vertex carrying the label i in the polygon. The last two elements contain pointers which point towards the nearest neighbour vertices of i in the list.

This storage of the polygon as an unordered list makes the implementation of the BFACF algorithm very effective. We note that BFACF moves of type II correspond to a change in the coordinates of only one vertex in the list. Moves of type I are implemented by either adding vertices to or deleting vertices from the list. If the list contains n vertices, then the addition of two vertices to this list in a BFACF move of type I is performed by adding the new vertices in the $(n+1)$th and $(n+2)$th addresses. Removal of two vertices from the list is accomplished by moving the last two vertices in the list to the newly opened addresses. In this way, the number of operations per attempted BFACF move is O(1), independent of the length of the polygon. At each step of the calculation we check that the polygon is self-avoiding using hash-coding (Knuth 1973). For a more detailed explanation of the hash-table, especially with respect to the simulation of walks, see the paper by Madras and Sokal (1988).

Unfortunately, the BFACF algorithm suffers from a very long autocorrelation time. The exponential autocorrelation time of this algorithm (which controls the relaxation of the Markov chain from an initial configuration to equilibrium) is infinite (Sokal and Thomas 1988). The integrated autocorrelation time, which controls the statistical

error in measured quantities, is also very long. This is unfortunate for the implementation here, since this algorithm is the only one known to be ergodic for a polygon of fixed knot type. The situation can be improved, to some extent, by the introduction of extra Monte Carlo moves in the BFACF algorithm. These moves are borrowed from the pivot algorithm (Carracciolo *et al* 1990), and have been shown to reduce the autocorrelation time of the slowest modes in the BFACF algorithm. In the case of polygons, we can use additional moves from the pivot algorithm for polygons (Dubins *et al* 1988, Madras *et al* 1990, Janse van Rensburg *et al* 1990), which has been well studied in the past few years.

We perform a pivot on the polygon in the following way. Choose two vertices on the polygon with uniform probability. These vertices (t_1 and t_2, say) seperate the polygon into two disjoint segments; let the shorter of the two segments by ω_1. The elementary transitions of the pivot algorithm are operations of elements of the symmetry group of the lattice (the octahedral group), on ω_1, such that t_1 and t_2 are left unchanged, or are reflected or rotated into each other. The possible transitions depend on the relative positions of t_1 and t_2. It is always possible to perform an inversion through the centre of mass of t_1 and t_2, so that there is at least one transition which can be attempted for all locations of the pivots. We implemented the pivot algorithm with inversions, 90° rotations around lattice axis, and reflections through planes which contain one lattice axis, and are inclined at 45° to the other two lattice axes, as well as through planes which contain two of the three lattice axes. With these choices of elementary transitions the pivot algorithm is ergodic for polygons of fixed length in the cubic lattice (Madras *et al* 1990). If we add these transitions from the pivot algorithm for polygons to the BFACF algorithm, then the algorithm is ergodic in the set of all polygons in three dimensions. The knot types of the polygons are now connected by transitions from the pivot algorithm. The philosophy of the implementation of the BFACF algorithm with pivots is to attempt a pivot transition after every m attempted BFACF move. (When we study polygons with fixed knot type, we have to check that the knot type does not change under these moves. We discuss this later.)

The fact that the implementation of the pivot algorithm here is ergodic for polygons of fixed length is useful to us. It is now possible to test the BFACF algorithm by using it to calculate the mean square radius of gyration of polygons of fixed size. We do this by running the algorithm, and calculating the square radius of gyration very time that the current polygon has the desired length. The results obtained can be compared with results from the pivot algorithm at fixed n. A run of the BFACF/pivot algorithm of 625 000 000 attempted BFACF moves, and one attempted pivot every 125 attempted BFACF moves, gives for the mean square radius of gyration $\langle r^2(n)\rangle$ of polygons with n edges:

$$\langle r^2(100)\rangle = 22.58 \qquad \langle r^2(200)\rangle = 51.06. \tag{3.9}$$

If we run the pivot algorithm alone at fixed n, then runs of 200 000 iterations at $n = 100$ and 100 000 at $n = 200$ gives

$$\langle r^2(100)\rangle = 22.66 \pm 0.22 \qquad \langle r^2(200)\rangle = 51.69 \pm 0.72. \tag{3.10}$$

where the error bars are 95% confidence intervals.

The most effective implementation of the BFACF/pivot algorithm occurs when the ratio of attempted BFACF moves to attempted pivot moves gives the optimum conditions for estimating variables that we are measuring by using the algorithm. A suitable

measure for the effectiveness of Monte Carlo simualtions was introduced by Hammersley and Handscomb (1967). The *relative efficiency* of a Monte Carlo method with respect to the observable ζ, $\chi(\zeta)$, is defined by

$$\chi(\zeta) = T\sigma^2 \tag{3.11}$$

where T is the CPU time taken by the algorithm, and σ is the variance in the mean of the quantity ζ. In applying equation (3.11) we shall express σ as a fraction of the estimate of the observable ζ. An algorithm which is efficient in calculating an estimate of a given quantity will have a lower value of χ than an algorithm which is less efficient at estimating the quantity.

We performed a series of runs to determine the optimum ratio of BFACF moves to pivots. The results are listed in table 1. In the first column we list the number of iterations (we take one iteration to be 5000 attempted BFACF moves). The second column lists the number of attempted BFACF moves for every attempted pivot. The third column lists the CPU time (in seconds) of the run. We measured χ with respect to three observables: the mean length of the polygons, the mean square radius of gyration of the polygons, and the exponent ν calculated from the data. Over all these runs the mean length of the polygons were 246 ± 11, the mean square radius of gyration was 67.9 ± 4.2, and the exponent ν was calculated to be 0.590 ± 0.004. The relative efficiencies in calculating these numbers are listed in the next three columns of table 1. There is a clear minimum when the ratio of attempted BFACF moves to pivots is near 64.

Table 1. The relative efficiency of the BFACF/pivot algorithm for polygons measured with respect to $|\omega|$, $\langle r^2 \rangle$ and ν. The knot-type of the polygon was not examined after a successful pivot.

Iterations	BFACF/pivot	CPU	$\chi(\lvert\omega\rvert)$	$\chi(r^2)$	$\chi(\nu)$
100 000	8192	57 400	130	189	1246
50 000	2048	29 600	104	156	1186
50 000	512	30 000	83	105	565
50 000	128	34 900	64	87	583
50 000	64	39 600	45	68	397
50 000	32	49 100	56	85	558
30 000	8	70 300	123	186	1374

In this paper we apply the BFACF/pivot algorithm to polygons of fixed knot type. Since the pivot algorithm may change the knot-type of a polygon, we must examine the polygon after every successful pivot to make sure that it is still the desired knot type. We do this by calculating the Alexander polynomial of the polygon (see e.g. Vologodskii *et al* 1974, Frank-Kamenetskii *et al* 1975, Michels and Wiegel 1984, 1986, Janse van Rensburg and Whittington 1990) and reject the move if the Alexander polynomial changes. Including this calculation makes it more expensive to perform pivots so that we expect a change in the relative efficiency of the algorithm, which we illustrate in table 2. There is a broad minimum in this table where the ratio BFACF/pivot is between 64 and 512. As our optimal choice, we pick this ratio to be 125 in our subsequent applications.

Table 2. The relative efficiency of the BFACF/pivot algorithm for polygons measured with respect to $|\omega|$, $\langle r^2 \rangle$ and ν. In this case we check the knot-type of the polygon after every successful pivot.

Iterations	BFACF/pivot	CPU	$\chi(\|\omega\|)$	$\chi(r^2)$	$\chi(\nu)$
100 000	8192	57 400	130	189	1246
50 000	2048	30 400	107	160	1218
50 000	512	33 600	93	117	632
50 000	128	50 300	92	126	841
50 000	64	70 600	80	121	706
50 000	32	110 900	126	192	1260
30 000	8	218 900	383	579	4276

4. Numerical results

4.1. The exponent $\nu_{\mathcal{K}}$

We expect one length scale in this problem (for each fixed knot type). This implies that there is a single exponent $\nu_{\mathcal{K}}$ (which may depend on the knot type) which characterizes the divergence of all metric quantities in the model. We defined two such quantities, the mean square radius of gyration and the mean span, in equations (3.3) and (3.4). In this section we examine the exponent $\nu_{\mathcal{K}}$ for polygons of fixed knot type; we are interested in the proposition that the value of this exponent may depend on the knot type.

We simulated polygons for $2\times5\times10^5$ iterations, where one iteration is 2500 attempted BFACF transitions with an attempted pivot every 125 attempted BFACF moves. In total, the number of BFACF moves attempted was $6\times25\times10^8$, and the total number of attempted pivots was 5×10^6. The mean square radius of gyration, and the mean span were calculated and recorded after every iteration. We calculated autocorrelation times as set out in section 2 and used equation (2.6) to estimate error bars on the quantities calculated in these runs. The value of $n_{\min}$ in equation (3.7) was set to 150, and $n_{\max}$ was taken to be 800 for the unknot and the trefoil, and 900 for the figure eight, the double trefoil and the knot 6_1 (see for example Burde and Zieschang 1985).

The expected dependence of the mean square radius of gyration and the mean span on the length of the polygons as defined in equations (3.3) and (3.4) is only asymptotic. In general, we expect corrections to scaling in these expressions (Le Guillou and Zinn-Justin 1980, 1989). The corrections to scaling are usually dominated by a confluent correction, so we expect that the mean square radius of gyration data can be adequately described by

$$\langle r^2 \rangle = (C_{\mathcal{K}} + b'_{\mathcal{K}} n^{-\Delta} + c'_{\mathcal{K}} n^{-1}) n^{2\nu_{\mathcal{K}}}. \tag{4.1}$$

The exponent Δ takes care of confluent corrections to scaling and its value is close to 0.5 (Le Guillou and Zinn-Justin 1980, 1989). The term which contains an inverse power of n is the usual analytic correction. The same behaviour is expected for the mean span. We expect that

$$\langle s \rangle = (D_{\mathcal{K}} + u'_{\mathcal{K}} n^{-\Delta} + v'_{\mathcal{K}} n^{-1}) n^{\nu_{\mathcal{K}}}. \tag{4.2}$$

If we take logarithms on both sides of equations (4.1) and (4.2), and expand the logarithm, then we find that we can determine the exponent from a three-parameter

linear fit where we take the possible confluent corrections to scaling into account. We find

$$\log\langle r^2\rangle = \log C_{\mathscr{K}} + b_{\mathscr{K}} n^{-\Delta} + 2\nu_{\mathscr{K}} \log n \tag{4.3}$$

$$\log\langle s\rangle = \log D\mathscr{K} + u_{\mathscr{K}} n^{-\Delta} + \nu_{\mathscr{K}} \log n. \tag{4.4}$$

We examined data for five different knots: the unknot (O), the trefoil (T), the double trefoil (D), the figure eight knot (F) and the knot 6_1 (S). Our best estimates for $\nu_{\mathscr{K}}$ from the mean square radius of gyration data are

$$\nu_{\mathrm{O}} = 0.596 \pm 0.021 \tag{4.5}$$

$$\nu_{\mathrm{T}} = 0.599 \pm 0.086 \tag{4.6}$$

$$\nu_{\mathrm{D}} = 0.641 \pm 0.078 \tag{4.7}$$

$$\nu_{\mathrm{F}} = 0.638 \pm 0.92 \tag{4.8}$$

$$\nu_{\mathrm{S}} = 0.64 \pm 0.11 \tag{4.9}$$

where the error bars are 66% confidence intervals. We see that, within the confidence interval, all these numbers include the exponent of the unknotted polygon (4.5). The results from the mean span data give better estimates for $\nu_{\mathscr{K}}$ (with smaller error bars). We find

$$\nu_{\mathrm{O}} = 0.5750 \pm 0.0077 \tag{4.10}$$

$$\nu_{\mathrm{T}} = 0.589 \pm 0.034 \tag{4.11}$$

$$\nu_{\mathrm{D}} = 0.607 \pm 0.037 \tag{4.12}$$

$$\nu_{\mathrm{F}} = 0.615 \pm 0.031 \tag{4.13}$$

$$\nu_{\mathrm{S}} = 0.597 \pm 0.044. \tag{4.14}$$

All these results include the 'exact' value of ν, expected to be near 0.588. The exponent seems to be independent of knot type. It may be that most knots are quite local objects in the polygon so that, in the language of the renormalization group, we can think of these knots being 'renormalized' away, so that they will have no effect on the exponents of the polygon.

4.2. The amplitude of knotted polygons

A knot is a (topological) constraint on the polygon. The results in the previous section indicate that this constraint (unlike the self-avoiding condition) does not have an effect on the critical exponent ν of the polygon. If 'universality' is preserved in this way, then the next level where the knot can make its presence felt is in the amplitudes, that is, in the values of the numbers $C_{\mathscr{K}}$ and $D_{\mathscr{K}}$ in equations (3.3) and (3.4).

To study this possibility, we first divide equations (4.3) and (4.4) by $n^{2\nu}$ and n^{ν} (we assume that ν is now independent of the knot type) to find that

$$\langle r^2\rangle / n^{2\nu} = C_{\mathscr{K}} + b_{\mathscr{K}} n^{-\Delta} + c_{\mathscr{K}} n^{-1} \tag{4.15}$$

$$\langle s\rangle / n^{\nu} = D_{\mathscr{K}} + u_{\mathscr{K}} n^{-\Delta} + v_{\mathscr{K}} n^{-1}. \tag{4.16}$$

To what extent are $C_{\mathscr{K}}$ and $D_{\mathscr{K}}$ dependent on $\mathscr{K}$? To answer this question, we first consider a few plots. In figure 2(*a*) we plot $\langle r^2\rangle / n^{2\nu}$ against $n^{-\Delta}$ for the unknot (O),

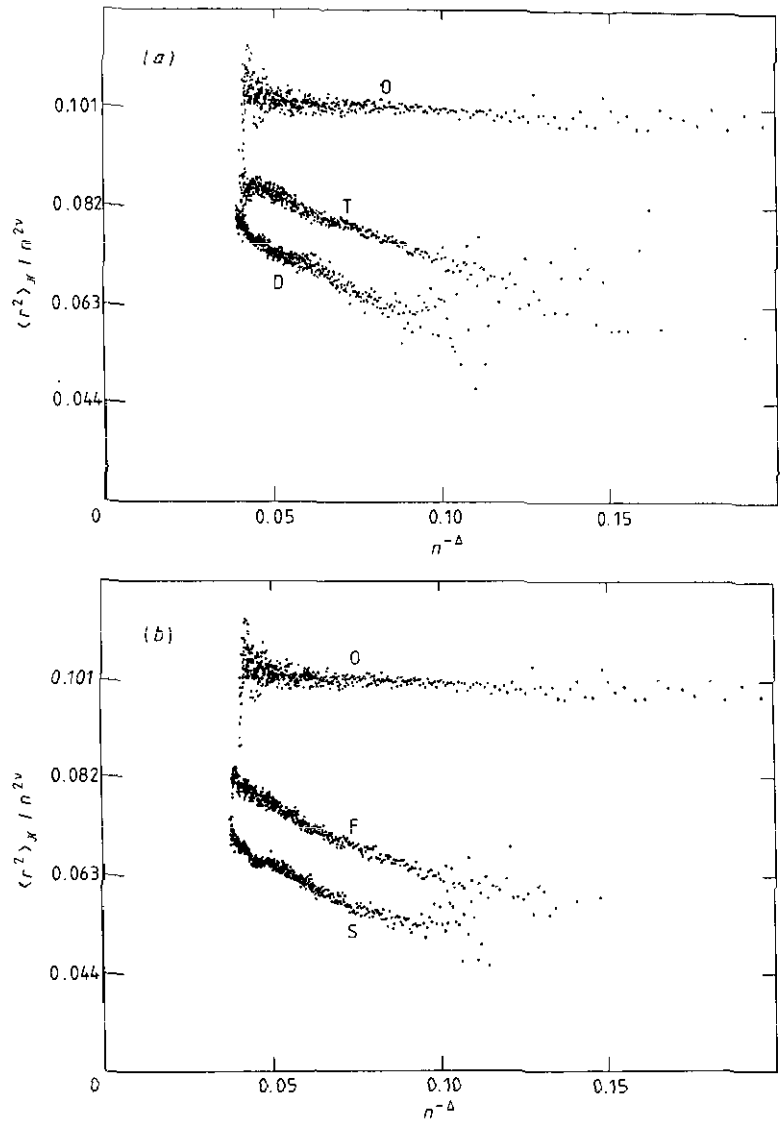

Figure 2. (*a*) $\langle r^2\rangle/n^{2\nu}$ against $n^{-\Delta}$ for the unknott (O), the trefoil (T) and the double trefoil (D). (*b*) $\langle r^2\rangle/n^{2\nu}$ against $n^{-\Delta}$ for the unknot (O), the figure eight knot (F) and the knot 6_1 (S).

the trefoil (T) and the double trefoil (D). Figure 2(b) is a similar plot containing the data for the unknot (O), the figure eight knot (F) and the knot 6_1 (S). The spread in the data points at large n is due to fewer data points being taken in that regime (near and above $n_{\max}$ in equation (3.7)). If we extrapolate the best straight lines throughout the data points in these two graphs, then the intercepts of each of these lines with the vertical axis are in each case near 0.10, *independent* of the knot type. Alternatively, we can plot the ratio $\langle r^2\rangle_{\mathrm{O}}/\langle r^2\rangle_{\mathcal{K}}$ against $n^{-\Delta}$. We do this in figure 3 where we pick $\mathcal{K}$ to be T, D and S. If we extrapolate the curves to the verical axis, then we see that the intercept is in each case close to 1. We interpret this as evidence that $C_{\mathcal{K}}$ is independent

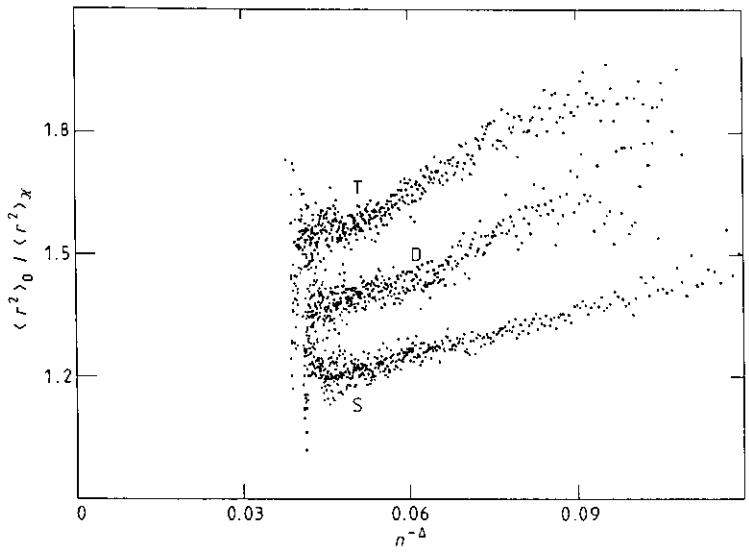

Figure 3. $\langle r^2\rangle_O/\langle r^2\rangle_{\mathscr{K}}$ against $n^{-\Delta}$ for $\mathscr{K}$ the trefoil (T), the double trefoil (D) and the knot 6_1 (S).

of $\mathscr{K}$ (in equation (4.15)), if we assume that ν and Δ have their field theory values. We note that we can determined $C_{\mathscr{K}}$ from equation (4.15) and our data via a three-parameter linear fit. If we do this with the data in figures 2(*a*) and 2(*b*), then we find as the best estimates for $C_{\mathscr{K}}$

$$C_O = 0.105 \pm 0.006 \tag{4.17}$$

$$C_T = 0.099 \pm 0.019 \tag{4.18}$$

$$C_D = 0.099 \pm 0.027 \tag{4.19}$$

$$C_F = 0.102 \pm 0.019 \tag{4.20}$$

$$C_S = 0.091 \pm 0.026. \tag{4.21}$$

We can examine the data for the mean span in the same way: A plot of $\langle s\rangle/n^{\nu}$ against $n^{-\Delta}$ is illustrated in figure 4 for each of the knot types. Again, we note that an extrapolation of linear curves thgough these data sets will intercept the vertical axis near 0.75. We can also plot $\langle s\rangle_O/\langle s\rangle_{\mathscr{K}}$ against $n^{-\Delta}$ (figure 5), where we take $\mathscr{K}$ to be T, D and S. The best straight lines through these data points intercepts the vertical axis again near 1; the mean span of the unknot is the same as that of any other knot in the scaling (large n) limit. This we interpret as evidence that $D_{\mathscr{K}}$ is not dependent on $\mathscr{K}$. A linear three-parameter fit to our data and equation (4.16) gives

$$D_O = 0.753 \pm 0.015 \tag{4.22}$$

$$D_T = 0.731 \pm 0.056 \tag{4.23}$$

$$D_D = 0.732 \pm 0.082 \tag{4.24}$$

$$D_F = 0.735 \pm 0.067 \tag{4.25}$$

$$D_S = 0.703 \pm 0.093. \tag{4.26}$$

These numbers are strong evidence that the amplitude of the mean square radius of gyration, and the mean span, are *independent* of the knot type of the polygon. This is

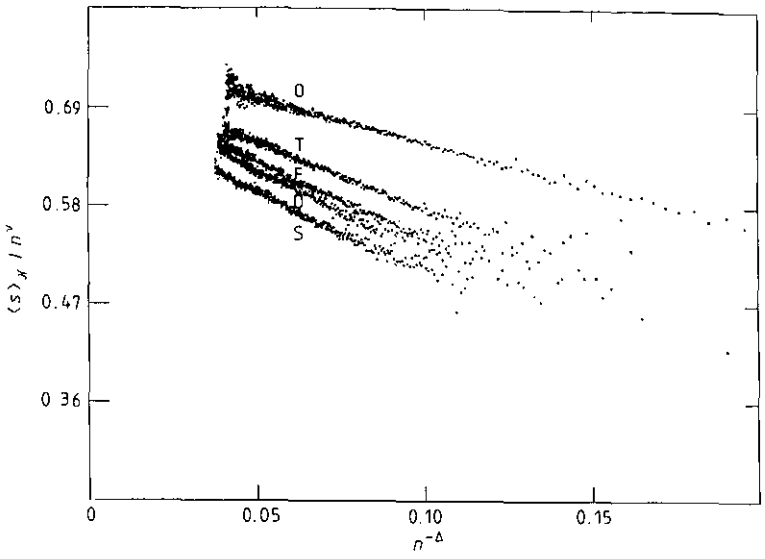

Figure 4. $\langle s \rangle / n^{\nu}$ against $n^{-\Delta}$ for the unknot (O), the trefoil (T), the figure eight knot (F), the double trefoil (D) and the knot 6_1 (S).

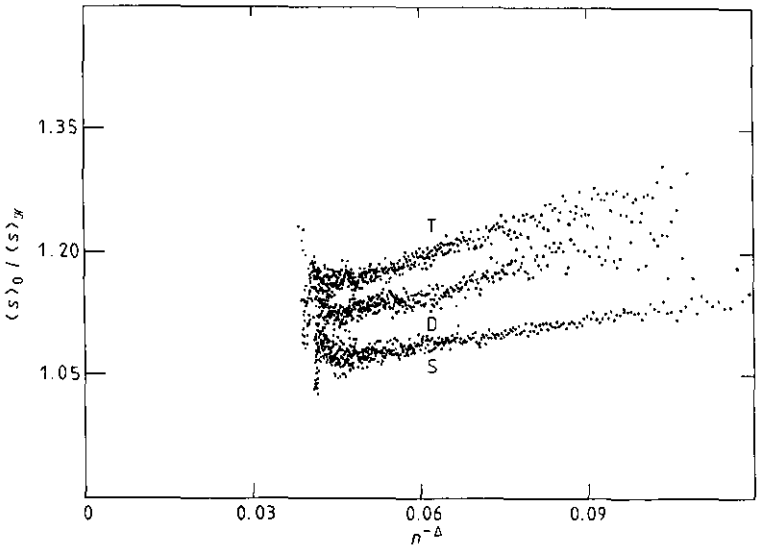

Figure 5. $\langle s \rangle_{O} / \langle s \rangle_{\mathcal{K}}$ against $n^{-\Delta}$ for $\mathcal{K}$ the trefoil (T), the double trefoil (D) and the knot 6_1 (S).

a remarkable conclusion. The knot type only affects corrections to scaling, that is, the rate of approach to the limiting behaviour.

5. Discussion

In this paper we have used a Monte Carlo algorithm which is a combination of the BFACF algorithm and the pivot algorithm to study the dimensions of polygons with

fixed knot type. The point of using the hybrid algorithm is that we want to generate polygons with a range of values of n in the sample (hence the grand canonical BFACF algorithm), but we want to avoid the long correlation times (hence the use of pivots). The BFACF algorithm is particularly suitable to a study of fixed knot type polygons since it is known that the underlying Markov chain has the knot types as its ergodic classes. When we introduce the pivot moves, the Markov chain is then ergodic on the polygons so, to sample polygons with fixed knot type, it is necessary to compute a knot invariant for every successful pivot move and reject the move if the invariant changes.

We have used this hybrid scheme to examine several different types of knot on the simple cubic lattice. Our results suggest that the critical exponent (ν) associated with the mean-square radius of gyration and with the span is independent of knot type and, moreover, the associated amplitudes are independent of knot type. We believe that this is a significant and surprising result. Of course, polygons with fixed n and different knot types have different average dimensions but, as $n \to \infty$, these differences manifest themselves only in the rate of approach to the asymptotic behaviour. It is important to notice that we are concerned with the situation in which the knot type is fixed and the number of edges in the polygon goes to infinity. If the knot type is not controlled then the probability that a polygon is knotted goes to unity as $n \to \infty$ and, for many measures of knot complexity, the complexity (of the knot) increases linearly in n (Soteros *et al* 1991). There are interesting questions about the dimensions of polygons whose knot complexity is increasing with n, but we do not address these here.

Acknowledgments

The authors acknowledge helpful discusssions with D W Sumners, N Madras and H A Lim. This work was supported by a grant from NSERC of Canada. This work was also partially supported by the US Department of Energy under contract No. DEFC05-85ER250000.

References

Aragao de Carvalho C and Caracciolo S 1983 *J. Physique* **44** 323
Aragao de Carvalho C, Caracciolo S and Frölich J 1983 *Nucl. Phys.* B **251** 209
Berg B and Foester D 1981 *Phys. Lett.* **106B** 323
Brower R 1991 unpublished
Burde G and Zieschang H 1985 *Knots* (Berlin: de Gruyter)
Caracciolo S, Pelissetto A and Sokal A D 1990 *J. Stat. Phys.* **60** 1
de Gennes P-G 1984 *Macromolecules* **17** 703
Dubins L E, Orlitsky A, Reeds J A and Shepp L A 1988 *IEEE Trans. Inform. Theory* **IT-34** 1509
Frank-Kamenetskii M D, Lukashin A V and Vologodskii A V 1975 *Nature* **258** 398
Guttmann A J 1989 *J. Phys. A: Math. Gen.* **22** 2807
Hammersley J M and Handscomb D C 1967 *Monte Carlo Methods* (London: Methuen)
Janse van Rensburg E J and Whittington S G 1990 *J. Phys. A: Math. Gen.* **23** 3573
Janse van Rensburg E J, Whittington S G and Madras N 1990 *J. Phys. A: Math. Gen.* **23** 1589
Kemeny J G and Snell J L 1976 *Finite Markov Chains* (Berlin: Springer)
Knuth D E 1973 *The Art of Computer Programming* vol 3 (Reading, MA: Addison-Wesley)
Le Guillou J C and Zinn-Justin J 1980 *Phys. Rev.* B **21** 3976
—— 1989 *J. Physique* **50** 1365
Madras N, Orlitsky A and Shepp L A 1990 *J. Stat. Phys.* **58** 159

Madras N and Sokal A D 1988 *J. Stat. Phys.* **50** 109
Metropolis N, Rosenbluth A W, Rosenbluth M N, Teller A H and Teller E 1953 *J. Chem. Phys.* **21** 1087
Michels J P J and Wiegel F W 1984 *Phys. Lett.* **90A** 381
—— 1986 *Proc. R. Soc.* A **403** 269
Pippenger N 1989 *Disc. Appl. Math.* **25** 273
Sokal A D and Thomas L E 1988 *J. Stat. Phys.* **50** 109
Soteros C E, Sumners D W and Whittington S G 1991 *Math. Proc. Camb. Phil. Soc.* to appear
Sumners D W and Whittington S G 1988 *J. Phys. A: Math. Gen.* **21** 1689
Vologodskii A V, Lukashin A V, Frank-Kamenetskii M D and Anshelevich 1974 *Sov. Phys.-JETP* **39** 1059
Wasserman S A and Cozzarelli N R 1986 *Science* **232** 951
Wasserman S A, Dungan J M and Cozzarelli N R 1985 *Science* **229** 171