The knot probability in lattice polygons

1990 J. Phys. A: Math. Gen. 23 3573

(http://iopscience.iop.org/0305-4470/23/15/028)

J. Phys. A: Math. Gen. 23 (1990) 3573–3590. Printed in the UK

The knot probability in lattice polygons

E J Janse van Rensburg and S G Whittington

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

Received 24 January 1990, in final form 16 March 1990

Abstract. The incidence of knots in lattice polygons in the face-centred cubic lattice is investigated numerically. We generate a sample of polygons using a pivot algorithm and detect knotted polygons by calculating the Alexander polynomial. If $p_n^0(\emptyset)$ is the probability that a polygon with n edges is the unknot, then it is known that $\limsup_{n\to\infty} p_n^0(\emptyset)^{1/n} = e^{-\alpha_0} < 1$. We find that $\alpha_0 = (7.6 \pm 0.9)\times 10^{-6}$. The effect of the solvent quality on $p_n^0(\emptyset)$ is considered. Our data show that the probability of a polygon being knotted increases rapidly as the quality of the solvent deteriorates.

1. Introduction

The occurrence of knots and links in closed linear polymers poses interesting questions to polymer physicists, chemists and knot theorists. The effects of knots and links on polymer networks were considered by Edwards (1967, 1968). de Gennes (1984) looked at tight knots in polymers and the effect which they have on long-time memory effects in melts of crystallisable linear polymers. Exciting new applications of ideas from algebraic topology and knot theory were made in the study of the topology of DNA molecules, particularly through the efforts of Sumners (1986, 1987a, 1990). The mechanism of enzyme action on DNA has been analysed in some cases by Sumners (1987b), Wasserman and Cozzarelli (1986) and Wasserman *et al* (1985).

There are interesting problems associated with the tabulation and counting of knots (Burde and Zieschang 1985). We can assign a measure of complexity to any given knot (such as the minimum number of crossings in a planar projection of the knot). Can we find the total number of knots of any given complexity? Traditionally, knots are tabulated according to their minimum number of crossings. If we ignore the fact that some knots are chiral, then there is only one knot with three crossings. This is the trefoil. There is also one knot with four crossings, the figure eight knot. Any two knots k_1 and k_2 can be composed to form a knot $k_1\#k_2$, which we call a *compound* knot. k_1 and k_2 are the *factors* of the knot $k_1\#k_2$. It is easily seen that the operation $\#$ is associative and commutative. If a knot cannot be written as the composition of two factors, then we call the knot *prime*. The trefoil and figure eight knots are both prime, but the granny knot is compound, since we can factor it into two trefoils.

Counting prime knots by the minimum number of crossings is a difficult problem that has recieved much attention in the literature. Ernst and Sumners (1987) proved that the number of prime knots grows at least exponentially with the minimum number of crossings. This is a problem in the statistics of knots, and in this paper we consider

0305-4470/90/153573+18$03.50

a similar problem. Given a lattice polygon with n edges embedded in some lattice $\Lambda \subset \mathcal{R}^3$, what is the probability that the polygon is knotted?

Vologodskii *et al* (1974) were the first to conduct a detailed numerical study in *statistical knot theory* (a term due to D W Sumners). Such a study typically consists of two parts. The first is the numerical simulation of the circular polymer molecules, modelled by polygons on the lattice (Hammersley 1961, Kesten 1963, 1964, Martin *et al* 1967). Since the incidence of knots is low in polygons, most studies tend to ignore the self-avoiding property of the polymers and focus on polymers with Gaussian statistics instead (Frank-Kamenetskii *et al* 1975, des Cloisseaux and Mehta 1979). Michels and Wiegel (1984, 1986) consider a closed-ring model of vanishing thickness as a model for circular polymers; the vertices are point masses and the edges are chemical bonds with a restoring harmonic force around some edge length a. The ring is then placed in a random heat field, which crumples it.

The invention of highly efficient Monte Carlo algorithms for the simulation of lattice polygons (Dubins *et al* 1988, Madras *et al* 1990) solved the problems encountered with the simulation of circular polymers. We (Janse van Rensburg *et al* 1990) proved that the pivot algorithm is ergodic and reversible in the face-centred cubic (FCC) lattice and performed numerical simulations which demonstrated the efficiency of this algorithm. There is an advantage in studying the incidence of knots in polygons in the FCC lattice: on the simple cubic lattice, we can make a polygon with 24 edges which is a trefoil while on the FCC we need only 16 edges (figure 1).

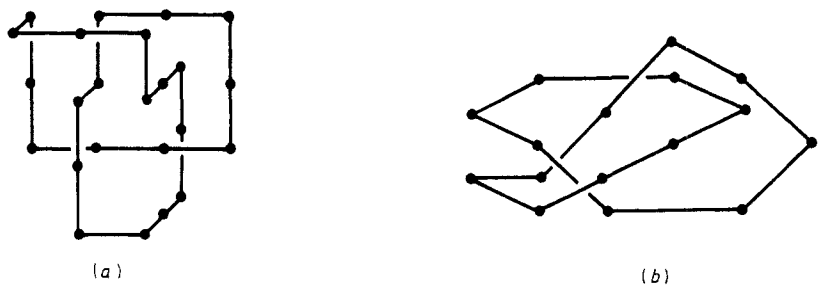

Figure 1. (a) A trefoil on the simple cubic lattice with 24 edges. (b) A trefoil on the FCC lattice with 16 edges.

The second part of a numerical simulation is the detection of knots through the calculation of an invariant. Vologodskii *et al* (1974), Frank-Kamenetskii *et al* (1975) and Michels and Wiegel (1984, 1986) all calculated the Alexander polynomial ($\Delta(t)$) at various values of t. We shall follow the same route since, as we shall point out, it is more efficient to calculate $\Delta(t)$ than, for example, the Jones polynomial.

The theoretical understanding of knotted walks and polygons came slowly. Kendall (1979) proved that every Brownian walk in three dimensions contains infinitely many disjoint 'knots' in every time interval. Sumners and Whittington (1988) and Pippenger (1989) proved independently that if $p_n^0(\emptyset)$ is the probability that a polygon in $\mathcal{Z}^3$ with n edges is the unknot ($\emptyset$), then $p_n^0(\emptyset)$ goes to zero exponentially fast with n as n tends to infinity: that is

$$\limsup_{n\to\infty} p_n^0(\emptyset)^{1/n} = e^{-\alpha_0} < 1. \tag{1.1}$$

Very little is known about the rate at which $p_n^0(\emptyset)$ goes to zero as $n \to \infty$. Preliminary results by Frank-Kamenetskii *et al* (1975) indicate that α_0 is very small. In this paper we shall calculate α_0 for the FCC lattice. We shall also consider the effect of the 'solvent' on α_0, which is important since it may influence the analysis of experiments by indicating the effect which a bad solvent will have on the knot probability of circular polymers.

In section 2 we briefly consider the pivot algorithm. We explain the extension of this algorithm to simulate polygons in a bad solvent and discuss strategies for improving it. In subsection 2.2 we demonstrate the effectiveness of the algorithm by calculating the heat capacity of the polygons.

In section 3 we consider the detection of knotted polygons. We discuss strategies which can be used to speed up this part of the simulation and consider their effectiveness, in particular the 'smoothing' of the polygons and the use of Reidemeister moves. In section 4 we present our numerical data. The knot probability of polygons in a good solvent is considered in subsection 4.1. We estimate α_0 in equation (1.1) and find $\alpha_0 = (7.6 \pm 0.9) \times 10^{-6}$ (where the error bar is one standard deviation). In subsection 4.2 we consider the effect of a bad solvent on the knot probability and we find that α_0 increases significantly as we decrease the quality of the solvent. These results imply that there is a correlation between the knot probability and the 'collapsed' state of the polygon. We demonstrate this correlation in subsection 4.3. Finally, we conclude the paper with a few remarks and observations in section 5.

2. The pivot algorithm and polygons

2.1. Ergodicity and reversibility

The pivot algorithm is a Monte Carlo algorithm which was invented by Lal (1969) for the simulation of self-avoiding walks in the canonical ensemble. The algorithm was studied in detail by Madras and Sokal (1988) who called it the pivot algorithm. Madras *et al* (1990) extended the pivot algorithm to polygons on the simple cubic lattice, extending a two-dimensional result of Dubins *et al* (1988). We (Janse van Rensburg *et al* 1990) subsequently studied the algorithm on the FCC lattice.

A *polygon* ω is defined as a sequence of lattice sites (vertices) $\omega_0, \omega_1, \omega_2, \ldots, \omega_n$, such that $\omega_0 = \omega_n$, ω_i and ω_{i+1} are neighbours in the lattice and the ω_i are distinct for all i, and the associated edges between pairs of vertices ω_i and ω_{i+1}. The basic elementary move of the algorithm is as follows. Choose two different vertices on ω (the current polygon) at random with (for example) a uniform distribution (this is sufficient, but not necessary), say ω_i and ω_j. With these *pivots*, apply an elementary transformation from a list of possible transformations to the shorter piece of the polygon (or alternatively, to the piece not containing the origin). If the result is a polygon (self-avoiding), then it is accepted, becoming the current polygon. Otherwise it is rejected, and the current polygon does not change. The list of possible elementary transformations is given by the octahedral group, which is the symmetry group of the cubic lattices. Among others, the elements of this group are point reflections (inversions), reflections through the lattice planes containing two of the three cartesian axes of the FCC, reflections through planes inclined at 45° to two cartesian axes and containing the third axis (45°-planes) and 90°-rotations about the cartesian axes. In

each case the 'origin' of the transformation is taken to be the midpoint of the line segment joining the two pivots. We (Janse van Rensburg *et al* 1990) proved the following theorem.

Theorem 2.1. The pivot algorithm applied to polygons on the FCC lattice is ergodic, provided that inversions, lattice plane reflections, and either 45°-plane reflections or 90°-rotations, have positive probability of occurring as elementary transformations in the Monte Carlo algorithm.

Let R_n be the set of all polygons with n vertices. We view the algorithm as a realisation of a Markov chain on R_n. Let us now consider reversibility and the simulation of a bad solvent. Let $R_n(m) \subset R_n$ be the set of all polygons with n vertices and m contacts (a 'contact' between two vertices occurs when two vertices are nearest neighbours on the lattice, but not nearest neighbours on the polygon). Let the cardinality of $R_n(m)$ be $p_n(m)$. We assign equal weights to the polygons in $R_n(m)$, but the polygons in $R_n(m)$ and $R_n(m')$ are not assigned equal weights. The effect of the solvent is taken into account by introducing an attractive monomer–monomer interaction between the vertices of the polygon (a repulsive monomer–solvent interaction could have been used instead since it only changes the partition function by a constant). So, for each $\omega \in R_n(m)$ we assign a weight $e^{m\phi}$, where ϕ is a reduced energy associated with the monomer–monomer interaction. We implement the simulation of polygons with this weight by using the Metropolis algorithm (Metropolis *et al* 1953). Then the algorithm realises a Markov chain with a state space R_n and an invariant probability measure

$$\pi_\omega(m) = \frac{e^{m\phi}}{\sum_{m=0}^{\infty} p_n(m) e^{m\phi}} \qquad \text{for each } \omega \in R_n \tag{2.1}$$

where ω has m contacts.

To see this, consider the following. Suppose that $\omega_m \in R_n(m)$ and $\nu_{m'} \in R_n(m')$. Let $\rho(\omega_m, \nu_{m'}, t)$ be the probability that $\nu_{m'}$ can be obtained from ω_m via the elementary transformation t with pivots r_1 and r_2 (for simplicity, suppose that we choose r_1 and r_2 uniformly on ω_m). Since the weight associated with ω_m is $e^{m\phi}$, and with $\nu_{m'}$ is $e^{m'\phi}$, then $\rho(\omega_m, \nu_{m'}, t)$ is given by

$$\rho(\omega_m, \nu_{m'}, t) = \frac{2}{n(n-1)} q(t; r_1, r_2) \begin{cases} e^{(m'-m)\phi} & \text{if } m' < m \\ 1 & \text{otherwise} \end{cases} \tag{2.2}$$

where $q(t; r_1, r_2)$ is the probability that we choose to perform the elementary transformation t given the pivots r_1 and r_2. Since t leaves r_1 and r_2 unchanged, it follows that

$$\rho(\omega_m, \nu_{m'}, t) = e^{(m'-m)\phi} \rho(\nu_{m'}, \omega_m, t^{-1}) \tag{2.3}$$

where t^{-1} is the inverse transformation of t. (Note that every elementary transformation, except for 90°-rotations, is its own inverse. To have reversibility, we must give 90°-rotations and −90°-rotations equal probability of occurring. Since the list of possible transformations depends only on the relative position of the pivots we have $q(t; r_1, r_2) = q(t^{-1}; r_1, r_2)$.) Summing over all r_1 and r_2, and all t which changes ω_m into $\nu_{m'}$, we find that

$$\rho(\omega_m \to \nu_{m'}) = e^{(m'-m)\phi} \rho(\nu_{m'} \to \omega_m) \tag{2.4}$$

where $\rho(\omega \rightarrow \nu)$ is the probability of obtaining ν from ω via any choice of pivots and any elementary tansformation.

The basic elementary transformation is described by a matrix $P = \{\rho(\omega \rightarrow \nu)\}$ which has the following properties.

(1) For each ω and ν in R_n there exists an $M \geq 0$ such that the M-step probability from ω to ν, $\rho^M(\omega \rightarrow \nu)$, is positive. This follows immediately from theorem 2.1 and equation (2.2). This is ergodicity.

(2) For each $\omega_m \in R_n(m)$, and for each m

$$\sum_{m'} \sum_{\nu_{m'} \in R_n(m')} \pi_\nu(m') \rho(\nu_{m'} \rightarrow \omega_m) = \pi_\omega(m) \tag{2.5}$$

which we easily verify from equation (2.1) and (2.4).

Therefore, π_ω is the unique limit distribution of the Markov chain on R_n with transition probability matrix P (Kemeny and Snell 1976).

We programmed this algorithm in FORTRAN 77 on an Apollo 10000 computer. We selected pivots with uniform probability and used inversions, lattice plane reflections, 45°-plane reflections and 90°-rotations as possible elementary transformations. For each choice of pivots, we picked an elementary transformation with uniform probability from a list of possible transformations. To compensate for bias in our simulation due to the initial configuration of the polygon we discarded the first 150 000 attempted transformations before we started observing the polygons. To check for self-intersections and to count the number of contacts in proposed polygons efficiently, we used hash-coding (Knuth 1973, Horowitz and Sahni 1976).

We calculated sample means using standard statistical techniques. Error bars were estimated by calculating the integrated autocorrelation function using the methods developed by Madras and Sokal (1988).

2.2. Heat capacity

In this section we consider briefly the effect of an increase in the contact potential ϕ on the properties of polygons. An increase in ϕ simulates an increasingly bad solvent, and it is believed that there exists a critical point ϕ_c where the walks or polygons will undergo a collapse transition and precipitate from the solution. This is known as the θ-point (McCrackin *et al* 1973, Rapaport 1974, Douglas and Freed 1985).

For the FCC lattice numerical estimates for ϕ_c vary significantly. McCrackin *et al* (1973) estimated a value near 0.13 while an exact enumeration study by Rapaport (1974, 1976) suggests that it is somewhere between 0.15 and 0.18. The difficulty in calculating a good value for ϕ_c stems from the fact that the apparent value of ϕ_c may be dependent on n, the length of the walks or polygons that we consider, and that it will converge only slowly to its infinite-n value.

The heat capacity of walks or polygons will be a good detector of a phase transition at ϕ_c. Let c_i be the number of contacts in the ith polygon in the MC simulation. Then the heat capacity is defined as

$$\mathcal{H}_n = \langle c_i^2 \rangle - \langle c_i \rangle^2. \tag{2.6}$$

We calculated $\mathcal{H}_n$ for $n = 200$, 400 and 800 with ϕ taking values between 0.0 and 0.15. An important feature of the algorithm is the acceptance fraction f of proposed transformations. The lower f, the longer we expect the autocorrelation times in our

simulation to be. For $n = 800$, $f = 0.24$ for $\phi = 0.0$, but it steadily decreases to $f = 0.074$ for $\phi = 0.10$ and $f = 0.016$ for $\phi = 0.15$. To compensate for the drop in f for larger ϕ we took observations in our simulation only every m attempted transformations, where we steadily increased m with increasing ϕ. We observed the number of contacts 50 000 times, each observation separated by m attempted transformations, $m = 10$ for $\phi = 0.0$ and $m = 250$ for $\phi = 0.15$ performing 12.5×10^6 iterations. We plotted $\mathcal{H}_n$ against ϕ as shown in figure 2.

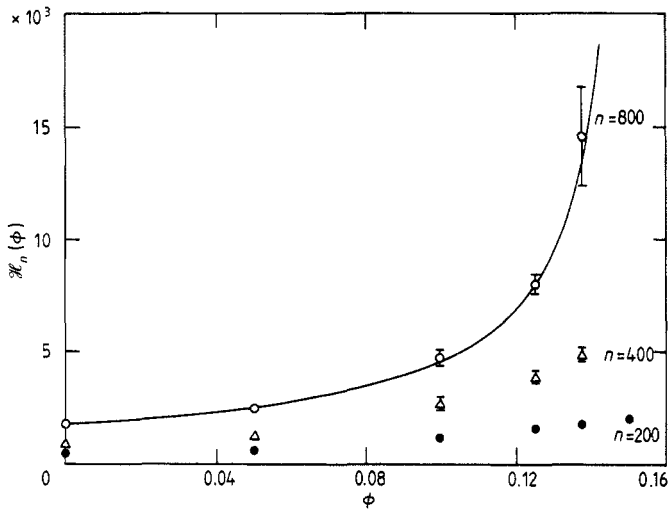

Figure 2. The heat capacity of polygons in a solvent with contact potential ϕ. The three sets of points corresponds to $n = 200$ (●), 400 (△) and 800 (○), where n is the number of edges in the polygon. For $n = 800$ it is apparent that there is a collapse transition near $\phi = 0.15$. The full curve is from a best fit to the data (equations (2.15) and (2.16)).

For finite n we do not expect a singularity in $\mathcal{H}_n$, but we expect that $\mathcal{H}_n$ will increase rapidly with ϕ near the critical point. For $n = 200$ and $n = 400$ there is not much sign of a transition for the range of ϕ values studied, but for $n = 800$ $\mathcal{H}_n$ increases rapidly as ϕ approaches a critical point near 0.15. If we assume that

$$\mathcal{H}_n(\phi) \approx C_n(\phi_c(n) - \phi)^{-\alpha} \tag{2.7}$$

where α is some effective exponent, then a best fit to the data will estimate the critical value of ϕ. We expect strong corrections to this scaling form, which will influence the value of α significantly. A least-squares fit to the data gives, for $n = 800$

$$\phi_c(800) = 0.153 \pm 0.005. \tag{2.8}$$

3. Detecting knots in polygons

In this section we consider the detection of knots in lattice polygons. We discuss methods for speeding up this process and study numerically the effectiveness of these methods.

3.1. The Alexander polynomial

The calculation of the Alexander polynomial for lattice polygons has been considered by Vologodskii *et al* (1974), Frank-Kamenetskii *et al* (1975) and Michels and Wiegel (1986). Let ω be a lattice polygon with n vertices. The first step in finding the Alexander polynomial $\Delta(t)$ at a given value of t is to find the Alexander matrix of a regular planar projection of ω. A regular projection of ω, $P(\omega)$ is obtained by first rotating ω in $\mathcal{R}^3$ by multiplication with an element of the rotation group SO(3) with irrational entries. This is to make sure that no two projected vertices of ω will fall on each other.

The next step is to search $P(\omega)$ for double points (which corresponds to over- and underpasses) and to tabulate them. We did this in a manner very similar to the Dowker–Thistlethwaite code (Dowker and Thistlethwaite 1983, Thistlethwaite 1985, Sumners 1987a). Let the chemical coordinate of a point $x \in P(\omega)$ be the distance, along the projection, of x from some arbitrary chosen origin. Suppose that $P(\omega)$ has l double points (crossings). Each crossing corresponds to an underpass of the projection. If we define a direction along $P(\omega)$ then each crossing is either positive or negative, as induced by the direction on the strands involved in the crossing. We illustrate this in figure 3. Record the following information about the crossings: the chemical coordinates of each underpass and overpass ($2l$ in all), and the sign of each crossing. We tabulate these data by increasing chemical coordinate of the underpasses. We call this table the *table of crossings* of the projection $P(\omega)$.

Figure 3. A + crossing (left) and a − crossing (right) in a projection.

It is now a simple task to construct the Alexander matrix from the information in the table of crossings (Fox 1962, Vologodskii *et al* 1974). The underpasses divides $P(\omega)$ into l arcs, and we label these arcs by integers 1 to l, the ith arc runs from the ith underpass to the $(i+1)$th underpass. The rows in the Alexander matrix correspond to the underpasses: Suppose that the jth arc overpasses the ith underpass, then the l entries in the i-row of the Alexander matrix are given by

$$a_{ik} = (\delta_{ij} + \delta_{i+1,j})(\delta_{i+1,k} + \delta_{ik}) + (1 - \delta_{ij})(1 - \delta_{i+1,j})f(t) \qquad \forall k = 1, 2, \ldots, l \quad (3.1)$$

and where $l+1$ is identified with 1. The function $f(t)$ is

$$f(t) = (\delta_{ik} - t\delta_{i+1,k} + (t+1)\delta_{kj}) \qquad \text{for + crossings} \quad (3.2)$$

and

$$f(t) = (-t\delta_{ik} + \delta_{i+1,k} + (t-1)\delta_{kj}) \qquad \text{for − crossings.} \quad (3.3)$$

The Alexander polynomial is the determinant of a minor of the Alexander matrix. The polynomial is not uniquely defined, it usually has a factor $\pm t^{\pm m}$, m being an arbitrary integer.

$\Delta(t)$ does not distinguish between all knots. An examination of a knot table (see e.g. Burde and Zieschang 1985) reveals that the knots 8_{20} and $3_1\#3_1$, 8_{21} and $3_1\#4_1$, and 8_{18} and $3_1\#3_1\#4_1$ have the same Alexander polynomials. In cases like these we always assumed that it is the knot with fewer crossings that we detected. We calculated $\Delta(-1)$ for all our polygons. If there is more than one knot which has the same value of $\Delta(-1)$, then we calculated $\Delta(-2)$ too. The calculation of $\Delta(-2)$ is slightly ambiguous, since the algorithm returns the number $\pm 2^{\pm m}\Delta(-2)$, m being some integer. To deal with this we constructed $(\pm 2^{\pm m}\Delta(-2))2^{\mp k}$, where we choose k to be the largest integer such that this product is an odd integer. In some rare cases there are knots for which $\Delta(-1)$ and $\Delta(-2)$ are the same, but $\Delta(-3)$ differs (for example 8_{10} and $3_1\#3_1\#3_1$). In those cases we calculated $\Delta(-3)$ too. In this way we could detect all knots up to seven crossings, and up to eight crossings if we ignore the few knots with degenerate Alexander polynomials.

The calculation of $\Delta(t)$ from the Alexander matrix takes $O(l^3)$ operations (to calculate the determinant), where l is the number of crossings in the projection of the polygon. In contrast to this, the Jones polynomial takes $O(2^l)$ operations (Kaufman 1989), and it is apparent that it will not be able to compete with $\Delta(t)$ in calculations of a statistical nature. Lattice polygons are not smooth objects, and we can expect numerous crossings in the projections, so that calculation of the Jones polynomial will be impracticable under these circumstances.

3.2. Smoothing the polygon

A polygon on the FCC lattice is a very irregular object with many sharp angles. This fact complicates the calculation of the knot type of a polygon, since a projection will contain many redundant crossings due to the irregular form of the polygon. It may therefore be advantageous to 'smooth' the polygon before we search for crossings in the projection.

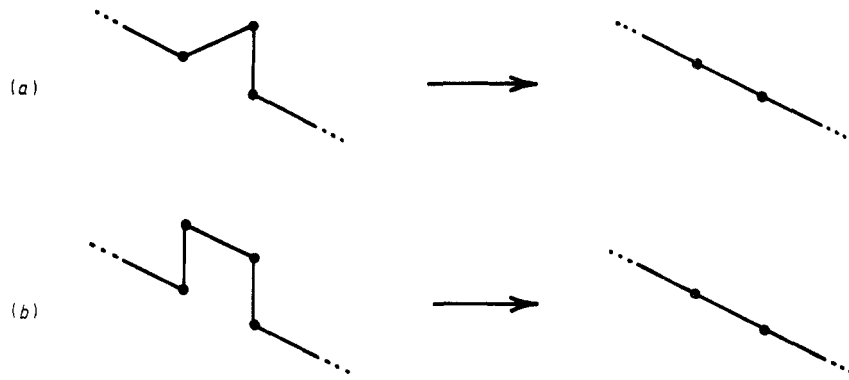

Figure 4. The two local moves used to 'smooth' the polygon.

One implementation of this idea is to snip redundant edges out of the polygon, simultaneously reducing the number of edges and making the polygon less irregular.

We used two local moves to perform this task. These are illustrated in figure 4. The search for locations on the polygon where we can apply the moves in figure 4 takes $O(n)$ operations, where n is the number of edges in the polygon. In contrast to this, to find the crossings in the projection takes $O(n^2)$ operations, since it involves comparing every two edges on the polygon. Smoothing is therefore very likely to be a successful strategy, since we perform $O(n)$ operations to reduce n before we perform a task taking $O(n^2)$ operations. We discuss the efficiency of smoothing in subsection 3.4.

3.3. Reidemeister moves

Reidemeister proved in the 1920s that two knots (or links) are ambient isotopic (can be deformed into one another without strand passage) if and only if their projections can be transformed into one another by a planar isotopy and three Reidemeister moves, illustrated in figure 5 (Reidemeister 1932). The projection of a knot may have many redundant crossings, and the application of the Reidemeister moves will reduce the number of such crossings. On a computational level, this reduction in the number of crossings should speed up the detection of knots.

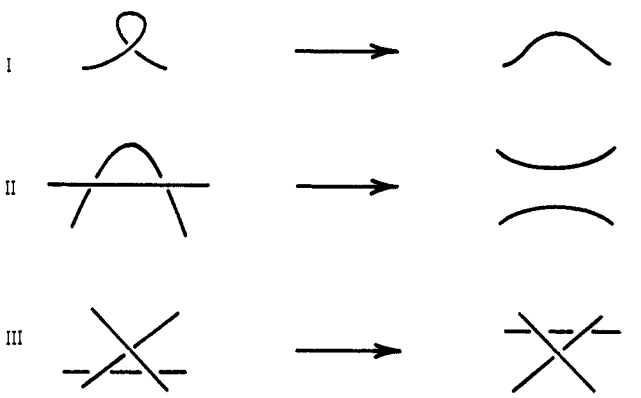

Figure 5. Reidemeister moves.

We explained the construction of a table of crossings for a projection in subsection 3.1. The advantage of this table is that we can perform the moves Reidemeister I and Reidemeister II directly in the table of crossings. The only extra calculation necessary is the ordering of the overpasses in increasing chemical coordinate, this takes $O(l \log_2 l)$ operations, where l is the number of crossings. This is because we have to search the arcs of the polygon between successive underpasses for overpasses to determine whether we can perform Reidemeister I or II.

We have already pointed out that the calculation of the Alexander polynomial takes $O(l^3)$ operations, it is therefore not obvious that for smallish l it will be of any benefit to perform these moves. In the case of the Jones polynomial, which takes $O(2^l)$ operations to construct, it may well be essential to perform these moves. In the next section we consider the effectiveness of smoothing and the Reidemeister moves in this calculation.

3.4. The effectiveness of smoothing and the Reidemeister moves

We start by summing up the ideas in subsections 3.2 and 3.3. In subsection 3.2 we argued that searching the polygon for crossings takes $O(n^2)$ operations, while performing the smoothing moves in figure 4 takes only $O(n)$ operations, where n is the number of edges in the polygon. If l is the number of crossings in the projection of the polygon, then calculating the Alexander polynomial takes $O(l^3)$ operations, while performing Reidemeister I and II takes $O(l \log_2 l)$ operations.

Table 1. CPU time (in seconds) taken by the algorithm to detect the knot type of 100 polygons of length $n = 800$. ϕ is the contact potential, and m is the number of attempted pivot transformations between every two observsations. The CPU times for the algorithm, with and without Reidemeister moves and smoothing, are recorded.

ϕ	m	None	Reid. I, II	Smoothed	Both
0	10	490	480	130	130
0.10	50	690	670	150	150
0.15	250	910	840	210	200

Since we expect that $n \gg l$ in most cases, most of the computing time will be taken by the search for crossings in the projection of the polygon. It is therefore to be expected that smoothing will be more efficient than the Reidemeister moves at speeding up the algorithm. To test this, we performed a series of short runs, recording the CPU time used by the algorithm to find the knot types of 100 polygons with $n = 800$ for $\phi = 0$, 0.10 and 0.15, the polygons are separated by 10, 50 and 250 attempted pivot transformations each. Our results are shown in table 1. m is the number of iterations performed between every two observed polygons.

Table 2. The effect of Reidemeister moves and smoothing on n, the number of edges in the polygon, and l, the number of crossings in the planar projection of the polygon. ϕ is the contact potential and m is the number of attempted pivot transformations between every two observations. The entries are in the format (n, l).

ϕ	m	None	Reid. I, II	Smoothed	Both
0	10	(800, 265)	(800, 55)	(390, 80)	(390, 10)
0.10	50	(800, 400)	(800, 135)	(365, 105)	(365, 25)
0.15	250	(800, 555)	(800, 230)	(315, 115)	(315, 35)

The column marked 'None' records the CPU time of the calculation without smoothing or Reidemeister I and II. We then switch the Reidemeister moves on, then smoothing, and then both. The increase in computing time with ϕ is due to the increase in m, the number of attempted pivot iterations between every two observed polygons. The times are recorded to the nearest 10 s. We notice that the application of Reidemeister moves I and II does not bring significant benefits, while smoothing reduces the CPU time by as much as 75%. These results do not imply that the application of Reidemeister moves is useless. To see their real value, let us consider the effect of the Reidemeister moves and smoothing on the number of edges (n) and the the number of crossings (l) in the projection of a polygon. We recorded the number of edges and the number of crossings for the short runs in table 1 and listed these,

to the nearest 5, in table 2, which has the same format as table 1. The entries in table 2 have the form (n, l). The Reidemeister moves reduce the number of crossings by more than 50% in each case, but our data show that smoothing performs better than the Reidemeister moves: It reduces n by about 50% *and* reduces the number of crossings more than the Reidemeister moves do for larger values of the contact potential ϕ. Smoothing becomes more effective with increasing ϕ, which is to be expected. The larger ϕ, the more crumpled we expect the polygons to be, so there will be more locations where we can apply the smoothing moves.

While the Reidemeister moves are not very effective in the calculation of the Alexander polynomial, they will be essential in the calculation of the Jones polynomial. We see that if we apply both smoothing and the Reidemeister moves, then the number of crossings is reduced to a level where calculation of the Jones polynomial may become possible, since (especially for the case $\phi = 0$) the numbers of crossings are low enough to consider this possibility. We only used the two moves in figure 4 in our smoothing subroutine. It is a simple matter to add additional moves to this list, or to imagine alternative schemes to extend this concept to make it more efficient.

4. Numerical results

In this section we present our numerical data on the incidence on knots in lattice polygons on the FCC lattice. We run the algorithm for mN attempted pivot transformations, where we determine the knot type of each polygon N times, each observation separated by m attempted transformations. Typically, we chose $N = 50\,000$ (or less in some cases), and m was adjusted to compensate for the decrease in the acceptance rate with increasing ϕ. We recorded the knot type and the number of contacts for each polygon. The most complicated prime knot which we found was 10_{36} and the most complicated compound knots were $3_1\#4_1\#5_2$ and $3_1\#10_{36}$.

To present fair error bars on our estimates, it was necessary to calculate the autocorrelation times in our MC simulations. This was done using the techniques developed by Madras and Sokal (1988). In subsection 4.1 we consider the knottedness of polygons in a good solvent, and we consider the effects of a bad solvent in subsection 4.2. The results in subsection 4.2 suggest that there is a strong correlation between the 'state of collapse' of a polygon and the probability that it is knotted. We explore this in subsection 4.3.

4.1. The incidence of knots in a good solvent

Vologodskii *et al* (1974) reported that the probability of finding a knot in a lattice polygon is very small for moderate values of n (about 100 edges): of the order 10^{-3} to 10^{-5}. In contrast to this, it is known that a lattice polygon will be a knot with probability 1 in the limit $n \to \infty$. In fact, the probability that a polygon is the unknot goes to zero exponentially fast as $n \to \infty$. If $p_n^0(\emptyset)$ is the probability that a polygon with n edges is the unknot ($\emptyset$), then Sumners and Whittington (1988) and, independently, Pippenger (1989), proved the following theorem.

Theorem 4.1.

$$\limsup_{n\to\infty} \left(p_n^0(\emptyset)\right)^{1/n} = e^{-\alpha_0} < 1$$

where α_0 is a positive constant.

Table 3. The knot probability $p_n^0(K)$ in a good solvent. The number of edges n was varied from 200 to 1600 and the contact potential $\phi = 0$. The autocorrelations τ_K are given in units of m iterations, where m is the number of attempted transitions between every two observations.

n	N	3_1	4_1	Total	$p_n^0(K)$	τ_K
200	50000	19	0	19	$(3.8 \pm 1.6) \times 10^{-4}$	1.08 ± 0.02
400	50000	118	3	121	$(2.42 \pm 0.44) \times 10^{-3}$	2.04 ± 0.06
800	50000	202	10	213	$(4.26 \pm 0.69) \times 10^{-3}$	2.8 ± 0.1
1200	25000	227	3	234	$(9.4 \pm 2.6) \times 10^{-3}$	9.2 ± 1.1
1600	20000	225	14	239	$(1.20 \pm 0.27) \times 10^{-2}$	6.2 ± 0.7

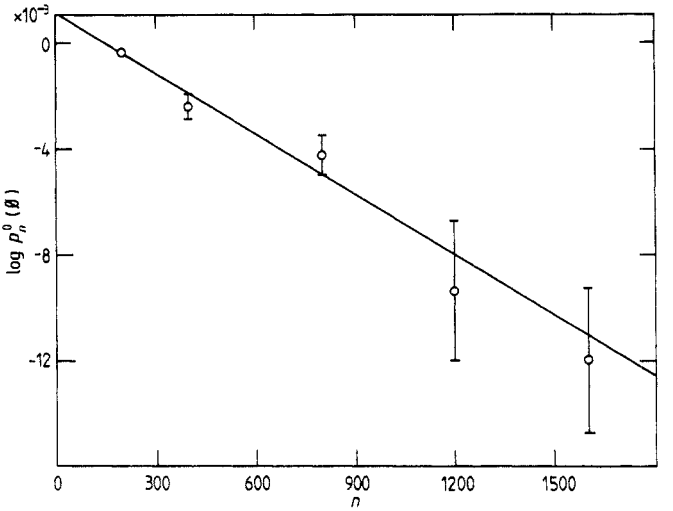

Figure 6. Plot of $\log p_n^0(\emptyset)$ against n. The full line is from the best fit to equations (4.1) and (4.2).

Let $p_n^0(K) = 1 - p_n^0(\emptyset)$ be the probability that a polygon with n edges is a knot. We calculated $p_n^0(K)$ at $\phi = 0$ for $n = 200$, 400, 800, 1200 and 1600. Our data are displayed in table 3 and plotted in figure 6. In each case we separated the observations by 10 attempted pivot transformations. A plot of $\log p_n^0(\emptyset)$ against n is linear, which suggests that

$$p_n^0(\emptyset) = C_0 e^{-\alpha_0 n} \tag{4.1}$$

where α_0 and C_0 are positive constants. A linear least-squares fit to the data in table 3 gives

$$\begin{aligned} \alpha_0 &= (7.6 \pm 0.9) \times 10^{-6} \\ C_0 &= 1.0011 \pm 0.0003 \\ \chi^2 &= 2.7 \\ \text{Prob}\,(\chi^2 \geq 2.7) &= 0.44 \end{aligned} \tag{4.2}$$

where χ^2 is the weighted mean-square deviation from the regression line. It should be distributed as $\chi^2(s-2)$, where s is the number of data points in the fit. Prob $(\chi^2 \geq 2.7)$

is the probability of obtaining a fit as poor as this, in this case 44%, so that we are confident that equation (4.1) is indeed satisfied by our data.

4.2. Knot probability in poor solvents

In this section we explore the effect of the solvent on the knot probability of polygons. We expect that a poor solvent will favour collapsed polygon configurations, so that it seems likely that the knot probability will increase with ϕ. Let $p_n^\phi(\emptyset)$ be the probability that a polygon with contact potential ϕ and n edges is the unknot. Let $p_n^\phi(K) = 1 - p_n^\phi(\emptyset)$. To investigate these probabilities we performed some runs at $n = 200$, 400 and 800 for values of ϕ varying between 0 and 0.15. The results for $n = 800$ are shown in table 4 and table 5. We list all the knots found in these runs in table 4, and in table 5 we list the knot probabilities and autocorrelation times for these runs. To compensate for the lower acceptance rate of the pivot algorithm for large ϕ, we increased the number of iterations between observations, m, from 10 to 250 as ϕ increased from 0 to 0.15. The number of observations, N, was typically 50 000. The autocorrelation times become long for higher values of ϕ.

Table 4. The knots found in the batch runs of 50 000 observations for polygons with $n = 800$ and ϕ between 0 and 0.15. The numbers of prime and compound knots are listed. The columns marked n_*, $n = 5, 6, 7$, are all the prime knots with n crossings.

ϕ	Prime	Composite	3_1	4_1	5_*	6_*	7_*	$3_1\#3_1$	$3_1\#4_1$
0	212	1	202	10	0	0	0	1	0
0.05	701	0	644	54	3	0	0	0	0
0.10	2 500	61	1 845	388	235	1	31	17	30
0.125	4 594	243	4 209	226	103	22	34	230	13
0.1375	11 162	1020	7 995	753	1981	62	147	399	130
0.15	18 713	2854	10 746	1404	4749	612	647	1163	261

Table 5. The knot probability $p_n^\phi(K)$ for polygons with $n = 800$ and for ϕ between 0 and 0.15. The autocorrelations τ_K are given in units of m iterations, where m is the number of attempted pivot transformations between every two observations.

ϕ	N	m	$p_n^\phi(K)$	τ_K
0	50 000	10	$(4.26 \pm 0.69) \times 10^{-3}$	2.8 ± 0.1
0.05	50 000	15	$(1.40 \pm 0.25) \times 10^{-2}$	11.2 ± 0.9
0.10	50 000	20	$(5.1 \pm 1.3) \times 10^{-2}$	80 ± 20
0.125	50 000	120	$(9.7 \pm 1.3) \times 10^{-2}$	44 ± 6
0.1375	50 000	175	$(2.43 \pm 0.53) \times 10^{-1}$	382 ± 170
0.15	70 000	250	$(4.31 \pm 0.50) \times 10^{-1}$	352 ± 130

We plotted $p_n^\phi(K)$ against ϕ for $n = 200$, 400 and 800, as in figure 7. For $n = 800$ we expect that there is a 'collapse transition' near 0.15, as we saw in subsection 2.2. We observe a dramatic rise in the knot probability as we approach this transition. We do not have any theoretical prediction such as theorem 4.1 for polygons in a bad solvent, but it is interesting to consider to what extent the linear behaviour in figure 7 survives as the solvent becomes increasingly poor. We therefore assume that

$$p_n^\phi(\emptyset) \approx C_\phi e^{-\alpha_\phi n}. \tag{4.3}$$

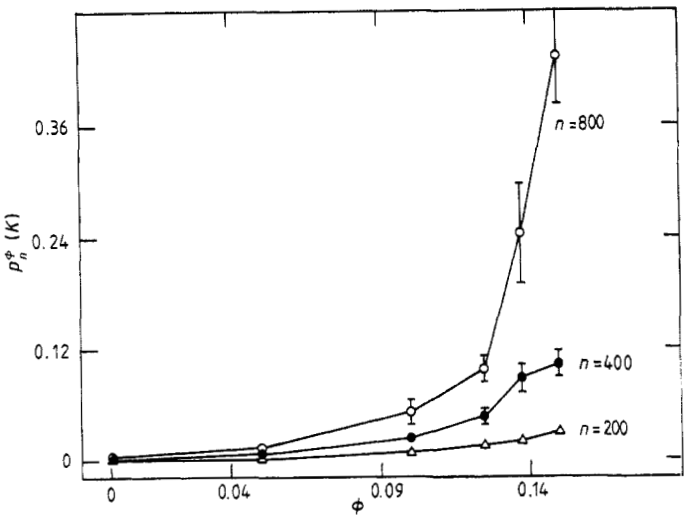

Figure 7. Plot of $\log p_n^\phi(K)$ against ϕ for $n = 200$ (△), 400 (●), and 800 (○). The full curves are drawn as a guide to the eye.

To explore this possibility we calculated $p_n^\phi(K)$ for $\phi = 0.10$ and 0.125 for values of n between 200 and 800. Our results are presented in table 6. We performed 50 000 observations separated by 50 attempted transformations for $\phi = 0.10$ and by 125 attempted transformations for $\phi = 0.125$ for each data point.

Table 6. The knot probability $p_n^\phi(K)$ for $\phi = 0.10$ and 0.125 for $n = 200$ to $n = 800$.

n	$p_n^{0.10}(K)$	$p_n^{0.125}(K)$
200	$(0.68 \pm 0.12)\times10^{-2}$	$(1.44 \pm 0.18)\times10^{-2}$
300	$(1.33 \pm 0.24)\times10^{-2}$	$(2.95 \pm 0.44)\times10^{-2}$
400	$(2.28 \pm 0.29)\times10^{-2}$	$(4.59 \pm 0.65)\times10^{-2}$
500	$(3.15 \pm 0.52)\times10^{-2}$	$(7.00 \pm 0.83)\times10^{-2}$
600	$(3.93 \pm 0.70)\times10^{-2}$	$(7.5 \pm 1.1)\times10^{-2}$
800	$(5.1 \pm 1.3)\times10^{-2}$	$(9.7 \pm 1.2)\times10^{-2}$

In figure 8, we plot $\log p_n^\phi(\emptyset)$ against n for $\phi = 0.10$ and 0.125. The linear nature of the plots supports the *ansatz* (4.3). A least-squares fit to the data gives

$$\begin{aligned} &\alpha_{0.10} \approx (8.6 \pm 1.0)\times10^{-5} \\ &C_{0.10} \approx 1.0112 \pm 0.0028 \\ &\chi^2 = 0.22 \\ &\text{Prob}\,(\chi^2 \geq 0.22) = 0.93 \end{aligned} \tag{4.4}$$

for $\phi = 0.10$ and

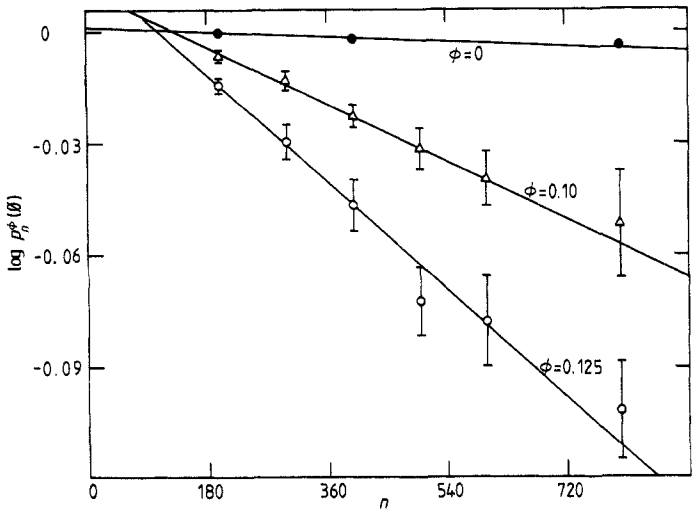

Figure 8. Plot of $\log p_n^\phi(\emptyset)$ against n for $\phi = 0$, 0.10 and 0.125. The full lines are the best fits to the data (4.3), (4.4) and (4.5).

$$\begin{aligned} &\alpha_{0.125} = (1.61 \pm 0.14) \times 10^{-4} \\ &C_{0.125} = 1.0177 \pm 0.0039 \\ &\chi^2 = 0.43 \\ &\text{Prob}\,(\chi^2 \geq 0.43) = 0.79 \end{aligned} \tag{4.5}$$

for $\phi = 0.125$. The numerical data provide excellent support for our ansatz (4.3). We see that the χ^2 values of both fits are very satisfactory. α_ϕ increases dramatically with increasing ϕ, so that the knot probability should be much higher in poor solvents than in good solvents. These results also suggest that there is a connection between the collapsed nature of the polygons and the probability that they are knots. We explore this connection in the next subsection.

4.3. Collapsed polygons and knots

We have not defined clearly what is meant by a 'collapsed polygon', and there are several possible definitions. In this subsection we say that the higher the number of contacts in a polygon, the more collapsed it is. This is a natural definition, for the more contacts, the more the polygon has excluded the solvent molecules from its immediate vicinity.

We have generated a large pool of data in calculating the knot probability of a polygon at fixed n for various values of ϕ. Associated with each polygon is a contact number and a knot type. In principle, we can therefore calculate the probability that a polygon with given contact number is a knot. To present smooth data, we calculated the knot probability of polygons with contact number in an interval $[c, c+\delta c)$ instead.

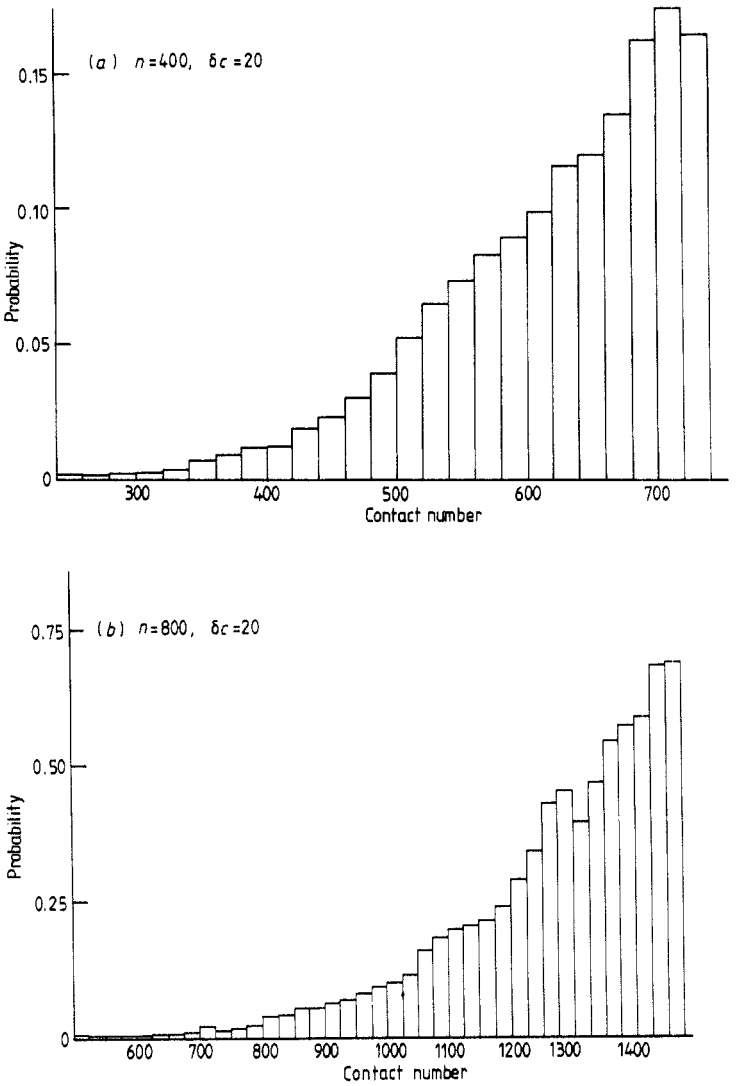

Figure 9. Histograms of the knot probability against the contact number, for (*a*) $n = 400$, $\delta c = 20$, and (*b*) $n = 800$, $\delta c = 25$.

In figure 9(*a*) we plotted this histogram for $n = 400$ and $\delta c = 20$, and in figure 9(*b*) for $n = 800$ and $\delta c = 25$. The knot probability is virtually zero for polygons with a low contact number, but rises rapidly once a threshold has been crossed, to as much as 0.20 for $c \approx 700$ and $n = 400$ and 0.70 for $c \approx 1400$ and $n = 800$.

5. Conclusions

In this paper we have accomplished three goals. The first was to develop an efficient numerical algorithm for the simulation of polygons and the detection of knots in polygons. The pivot algorithm proved successful in this regard, especially for small values of ϕ. For larger values of ϕ we were plagued by long autocorrelation times,

mainly due to the high rejection rate of proposed polygons in the pivot algorithm. This is partly compensated for by an increase in the rate at which polygons are being proposed. It is a strength of the pivot algorithm that it spends the least amount of time looking at polygons that will be rejected, and the longest time on those that will be accepted (Madras and Sokal 1988).

Secondly, we successfully introduced the idea of 'smoothing' into our algorithm. We saw that it brings a dramatic improvement in the performance of the algorithm. In addition to this, we supplemented the algorithm with two Reidemeister moves, which did not prove as effective as smoothing, but we pointed out that they may be essential in the calculation of alternative knot invariants, such as the Jones polynomial.

Our third accomplishment is the calculation of the knot probability of lattice polygons. We verified rigorous work in this regard (theorem 4.1) and showed that there is probably a relationship like equation (4.1) for polygons in bad solvents too. We also established a correlation between the knot probability of a polygon and the state of collapse: the more collapsed the polygon is, the more probable it is that it is a knot. In this sense, our results at large ϕ are completely compatible with the studies on polygons with Brownian statistics (Vologodskii *et al* 1974, Frank-Kamenetskii *et al* 1975, Michels and Wiegel 1986) where it is known that it is very likely for polygons to be knotted (see also the work of Kendall (1979) in this regard).

We conclude with a few remarks.

(1) Most of the CPU time in these calculations was spent looking for crossings in the projections of polygons. This part of the computation is ripe for vectorisation on a computer with parallel architecture. If implemented this will bring a major improvement in the performance of the algorithm with a resulting improvement in the error bars of the parameters that we calculated.

(2) It is believed that most compound knots are chiral. To test this prediction it is necessary to calculate an invariant which will detect chirality in knots, such as the Jones polynomial. The computation of this polynomial will take $O(2^l)$ operations if the projection of the knot has l crossings. We showed that the Reidemeister moves I and II can bring a major benefit to such a calculation, if it is supplemented by a procedure such as smoothing.

(3) By using methods similar to that of Sumners and Whittington (1988) and Pippenger (1989) we can show that there are exponentially few lattice polygons which are prime knots in the limit $n \to \infty$. Our data are not nearly good enough to study the rate at which knots turn compound with increasing n, though we can make a few observations. For $\phi = 0.125$ the fraction of compound knots increases from 0% at $n = 200$ to 3.2% at $n = 600$ and 5.0% at $n = 800$. There is also a similar increase in the fraction of compound knots with ϕ. For $n = 800$, the fraction of compound knots is 0% at $\phi = 0$, 5.0% at $\phi = 0.125$ and 13.0% at $\phi = 0.15$.

(4) We established a firm connection between contacts in polygons and the probability that the polygon is a knot. This has some significance in the analysis of DNA experiments. We now expect that a circular strand of DNA is more likely to be knotted if it is folded by an enzyme like topoisomerase (Sumners 1986) or if it is in a bad solvent. Water is a good solvent for DNA molecules, and our results indicate a low probability of knotting in a good solvent. The observation of knotted strands in the cell nucleus is therefore likely to be the result of the action of enzymes in the cell.

Acknowledgments

This research was financially supported by NSERC of Canada. We acknowledge helpful conversations with D W Sumners and N Madras.

References

Burde G and Zieschang H 1985 *Knots* (Berlin: de Gruyter)
de Gennes P-G 1984 *Macromolecules* **17** 703
des Cloiseaux J and Mehta M L 1979 *J. Physique* **40** 655
Douglas J F and Freed K F 1985 *Macromolecules* **18** 2445
Dowker C H and Thistlethwaite M B 1983 *Topology Appl.* **16** 19
Dubins L E, Orlitsky A, Reads J A and Shepp L A 1988 *IEEE Trans. Inform. Theory* **IT-34** 1509
Edwards S F 1967 *Proc. Phys. Soc.* **91** 513
—— 1968 *J. Phys. A: Gen. Phys.* Ser 2 **1** 15
Ernst C and Sumners D W 1987 *Math. Proc. Camb. Phil. Soc.* **22** 303
Fox R H 1962 *Topology of 3-manifolds and related topics* ed M K Fort (Englewood Cliffs, NJ: Prentice-Hall)
Frank-Kamenetskii M D, Lukashin A V and Vologodskii A V 1975 *Nature* **258** 398
Hammersley J M 1961 *Proc. Camb. Phil. Soc.* **57** 516
Horowitz E and Sahni S 1976 *Fundamentals of Data Structure* (Potomac, MD: Computer Science Press)
Janse van Rensburg E J, Whittington S G and Madras N 1990 *J. Phys. A: Math. Gen.* **23** 1589
Kaufman L H 1989 *Braid Group, Knot Theory and Statistical Mechanics* ed C N Yang and M L Ge (Singapore: World Scientific)
Kemeny J G and Snell J L 1976 *Finite Markov Chains* (Berlin: Springer)
Kendall W S 1979 *J. London Math. Soc.* **19** 378
Kesten H 1963 *J. Math. Phys.* **4** 960
—— 1964 *J. Math. Phys.* **5** 1128
Knuth D E 1973 *The Art of Computer Programming* **3** (Reading, MA: Addison-Wesley)
Lal M 1969 *Mol. Phys.* **17** 57
Madras N, Orlitsky A and Shepp L A 1990 *J. Stat. Phys.* **58** 159
Madras N and Sokal A D 1988 *J. Stat. Phys.* **47** 543
Martin J L, Sykes M F and Hioe F T 1967 *J. Chem. Phys.* **46** 3478
McCrackin F L, Mazur J and Guttman C M 1973 *Macromolecules* **6** 859
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
Rapaport D C 1974 *Macromolecules* **7** 64
—— 1976 *J. Phys. A: Math. Gen.* **9** 1521
Reidemeister K 1932 *Knotentheorie* (Berlin: Springer)
Sumners D W 1986 *Kem. Ind.* **35** 657
—— 1987a *J. Math. Chem.* **1** 1
—— 1987b *Geometry and Topology: Manifolds, Varieties and Knots* ed C McCrary and T Schifrin (New York: Dekker) p 297
—— 1990 *Math. Intell.* in press
Sumners D W and Whittington S G 1988 *J. Phys. A: Math. Gen.* **21** 1689
Thistlethwaite M B 1985 *London Math. Soc. Lect. Notes* **93** 1
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