Generating equilateral random polygons in confinement

Y. Diao[†], C. Ernst[*], A. Montemayor[*], and U. Ziegler[*]

[*]Department of Mathematics and Computer Science
Western Kentucky University
Bowling Green, KY 42101, USA
[†]Department of Mathematics and Statistics
University of North Carolina Charlotte
Charlotte, NC 28223

Abstract. One challenging problem in biology is to understand the mechanism of DNA packing in a confined volume such as a cell. It is known that confined circular DNA is often knotted and hence the topology of the extracted (and relaxed) circular DNA can be used as a probe of the DNA packing mechanism. However, in order to properly estimate the topological properties of the confined circular DNA structures using mathematical models, it is necessary to generate large ensembles of simulated closed chains (i.e., polygons) of equal edge lengths that are confined in a volume such as a sphere of certain fixed radius. Finding efficient algorithms that properly sample the space of such confined equilateral random polygons is a difficult problem. In this paper we propose a method that generates confined equilateral random polygons based on their probability distribution. This method requires the creation of a large database initially. However, once the database has been created, a confined equilateral random polygon of length n can be generated in linear time in terms of n. The errors introduced by the method can be controlled and reduced by the refinement of the database. Furthermore, our numerical simulations indicate that these errors are unbiased and tend to cancel each other in a long polygon.

AMS classification scheme numbers: Primary 57M25, secondary 92B99.

1. Introduction

This paper is motivated by the folding problem of circular DNA and other biopolymers confined to small volumes. It is known that DNA is highly condensed in all organisms. Thus the understanding of DNA folding under such conditions is a very important problem in biology and much effort has been devoted to this task. For instance, bacteriophages (viruses that infect bacteria) are commonly used to study DNA packing and folding in other dsDNA viruses because they have similar morphology and share similar assembly pathways, and the abundance of DNA knots found in the capsids provided a tool to probe the DNA packing inside the phage capsids [2, 3, 10, 11]. The geometry and topology of such condensed circular DNA is a difficult subject to study [8]. The commonly used models for circular DNA and biopolymers include off-lattice random polygons, the worm-like chain model, the beads model, and the random lattice polygon (walk) model. Of these models, random polygons are frequently used to model the behavior of polymers at thermodynamic equilibrium. The simplest and most fundamental type of random polygons are equilateral random polygons, namely polygons composed of freely jointed segments of equal length where the individual segments have no thickness. Such a random polygon is also known as an ideal random polygon and it is used to model the behavior of polymers under the so-called theta conditions where polymer segments that are not in direct contact neither attract nor repel each other. Much is known about the behavior of equilateral random polygons. For example, the overall dimensions such as the average end-to-end distance or the average radius of gyration is known to scale with the number of segments n as $\sqrt{n}$ [5, 6, 7]. However, in the case that the polygon is confined in a sphere, we still seem to lack a method that can effectively generate an equilateral random polygon according to its probability distribution.

The main subject of study in this paper is the generation of equilateral random polygons that are confined in a sphere of fixed radius. Our aim is to develop an algorithm that is capable of producing large sets of relatively long confined equilateral polygons according to their true probability distributions in a reasonable runtime. This is a known difficult problem. In the non-confined case, there are several known algorithms that work reasonably well. These include the crankshaft algorithm [9, 12], the hedgehog algorithm [9, 13] and the generalized hedgehog algorithm [15]. In the crankshaft algorithm, one starts from the regular n-gon (with unit edge length) in the plane. At any given step, two points in the polygon are selected at random. These two points define an axis that separates the polygon into two chains. One of the chains is selected at random and rotated around the axis by a random angle (the segments are allowed to cross each other in this process). The advantage of this method is that it has been shown to be ergodic [12]. That is, any possible configuration of an equilateral random polygon can be generated by this method. However, due to the high correlations of the edges generated this way, the above rotation process must be repeated $O(n)$ times to effectively eliminate any obvious correlation. Consequently, the run time needed for this algorithm

to generate an equilateral random polygon of length n is on the order of $O(n^2)$. The hedgehog algorithm, on the other hand, is faster, but it is unknown whether it is ergodic and also requires repeated rotation moves to get rid of the correlations among the initially generated random vector pairs. The generalized hedgehog algorithm, is the fastest (with a run time of $O(n)$ for generating an equilateral random polygon of length n), and is shown to be ergodic [15]. To generate an equilateral random polygon confined in a sphere of fixed radius, one could use the acceptance/rejection method based on one of the above methods. That is, one uses one of the above methods to generate non-confined polygons repeatedly but only keep those satisfying the confinement condition. For example, this approach was used in [1] in the investigation of the mean ACN of the confined equilateral random polygons. The main problem with this approach is the long time it requires when a large data set is to be generated, especially in the case that the polygons are long relative to the volume of the confinement since only exponentially few such polygons generated satisfy the confinement condition.

In this paper, we propose a fast algorithm that is capable of generating long equilateral random polygons confined within a sphere of fixed radius, thus providing a much needed tool for the study of confined DNA where this model is applicable. This method requires the creation of a large database initially. However, once the database is created, a confined equilateral random polygon of length n can be generated in linear time in terms of n. The errors introduced by the method can be controlled and reduced by the refinement of the database.

In the next section, we give some theoretical background for the algorithm. In Section 3, we introduce an algorithm for generating the equilateral random polygon with its theoretical probability distribution and describe how to modify this algorithm so that it can be used to generate an equilateral random polygon confined in a sphere with its theoretical probability distribution. In Section 4, we discuss how to numerically implement the algorithm. In Section 5 we present our numerical results with some discussions. These include a comparison study on the mean ACN with prior results and some discussions on the error estimations. We end our paper with remarks on a few possible directions for future study in Section 6.

2. The theoretical background of the algorithm

Let $U = (u, v, w)$ be a three-dimensional random vector that is uniformly distributed on the unit sphere S^2, i.e., the density function of U is

$$\varphi(U) = \begin{cases} \frac{1}{4\pi} & \text{if } |U| = \sqrt{u^2 + v^2 + w^2} = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Suppose U_1, U_2, ... , U_n are n independent random vectors uniformly distributed on S^2 (so the joint probability density function of the three coordinates of each U_j is simply $\frac{1}{4\pi}$ on the unit sphere). An equilateral random walk of n steps, denoted by

EW_n^O, is defined as the sequence of points in the three dimensional space $\mathbf{R}^3$: $X_0 = O$, $X_k = U_1 + U_2 + \cdots + U_k$, $k = 1, 2, ..., n$. Each X_k is called a vertex of the EW_n^O and the line segment joining X_k and X_{k+1} is called an edge of EW_n^O (which is of unit length). If the last vertex X_n of EW_n^O is fixed at X, then we have a conditioned random walk $EW_n^O|_{X_n=X}$. In particular, EW_n^O becomes a polygon EP_n if $X_n = O$. In this case, it is called an equilateral random polygon and is denoted by EP_n. The joint probability density function $f(X_1, X_2, ..., X_n)$ of the vertices of an EW_n^O is simply $f(X_1, X_2, ..., X_n) = \varphi(U_1)\varphi(U_2)\cdots\varphi(U_n) = \varphi(X_1)\varphi(X_2 - X_1)\cdots\varphi(X_n - X_{n-1})$ and the density function of X_k is given by

$$f_k(X_k) = \int\int\cdots\int \varphi(X_1)\varphi(X_2 - X_1)\cdots \varphi(X_k - X_{k-1})\mathrm{d}X_1\mathrm{d}X_2\cdots\mathrm{d}X_{k-1} \tag{2}$$

and it has the closed form [14]

$$f_k(X_k) = \frac{1}{2\pi^2 r_k}\int_0^\infty x\sin r_k x\left(\frac{\sin x}{x}\right)^k dx, \tag{3}$$

where $r_k = |X_k|$. Notice that $f_k(X_k)$ is in fact a function of r_k and k.

In theory, an equilateral random polygon of n edges can be generated one edge at a time in the following manner. Start at origin with $X_0 = O = X_n$, by symmetry, X_{n-1} is uniformly distributed on the unit sphere centered at O. Thus we choose X_{n-1} uniformly from this unit sphere. Once X_{n-1} is chosen, we are left with an equilateral random walk (of $n-1$ edges) with end points fixed at O and X_{n-1}. We can then choose X_{n-2} according to its distribution. Once X_{n-2} is chosen, we are left with an equilateral random walk (of $n-2$ edges) with end points fixed at O and X_{n-2}. We can then choose X_{n-3} according to its distribution and continue this process.

In practice, we immediately run into the following problem: In an equilateral random polygon with fixed end points at O and X_k ($k \geq 3$ since the case of $k \geq 2$ is rather obvious), what is the probability distribution of X_{k-1}? Theorem 1 provides an answer to this. However we need the following lemma first.

Lemma 1 *Let Y be uniformly distributed on the unit sphere centered at a fixed point $Y_0 \neq O$ and let $r = |Y|$. Then the probability density function $p(r)$ of r is given by $p(r) = \frac{r}{2r_0}$ where $r_0 = |Y_0|$ for $r \in [r_0 - 1, r_0 + 1]$ if $r_0 \geq 1$ and for $r \in [1 - r_0, 1 + r_0]$ if $r_0 < 1$.*

Proof. Without loss of generality, let us assume that Y_0 is on the y-axis so $Y_0 = (0, r_0, 0)$ as shown in Figure 1. Let θ be the angle between $\overrightarrow{Y_0Y}$ and the positive y-axis. We leave it to our reader to verify that $\cos\theta$ is uniformly distributed on $[-1, 1]$. Simple trigonometric calculations lead to

$$P(r \leq \tau) = P(\cos\theta \leq \frac{\tau^2 - 1 - r_0^2}{2r_0}) = \begin{cases} 1, & \tau > r_0 + 1; \\ \frac{1}{2}(1 + \frac{\tau^2 - 1 - r_0^2}{2r_0}), & |r_0 - 1| \leq \tau \leq r_0 + 1; \\ 0, & \tau < |r_0 - 1|. \end{cases}$$

It follows that

$$\frac{dP(r \le \tau)}{d\tau} = \begin{cases} \frac{\tau}{2r_0}, & |r_0 - 1| \le \tau \le r_0 + 1; \\ 0, & \text{otherwise.} \end{cases}$$

The result now follows.

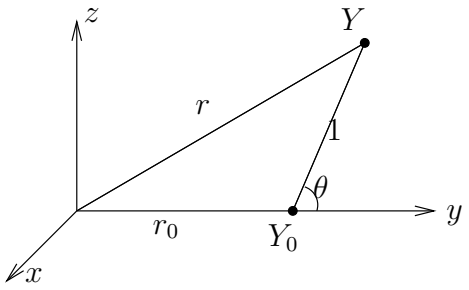


Figure 1. The determination of $p(r) = r/2r_0$.

Theorem 1 *Let EW_k^O be an equilateral random walk with k edges and $X_0, X_1, \ldots, X_k$ are its consecutive vertices. Let $r_j = |X_j|$ for $1 \le j \le k$. Let $g_j(r_j)$ be the probability density function of r_j. Then*

(a) $g_j(r_j) = 4\pi r_j^2 f_j(X_j)$ where f_j is given in (3);

(b) In case that r_{k+1} is fixed, then the conditional probability density function of r_k is given by

$$h_k(r_k|r_{k+1}) = \frac{r_{k+1}}{2r_k} \frac{g_k(r_k)}{g_{k+1}(r_{k+1})},$$

where $r_k \in [r_{k+1} - 1, r_{k+1} + 1]$ if $r_{k+1} \ge 1$ and $r_k \in [1 - r_{k+1}, 1 + r_{k+1}]$ if $r_{k+1} < 1$.

Proof. $\forall \delta > 0$ and $t > 0$, $s > 0$, consider the event $A_t : \ r_k \in [t - \delta, t]$ and $B_s : \ r_{k+1} \in [s - \delta, s]$. We have $P(A_t|B_s) = \frac{P(A_t \cap B_s)}{P(B_s)} = \frac{P(A_t)P(B_s|A_t)}{P(B_s)}$. Notice that $\lim_{\delta \to 0} P(A_t|B_s)/\delta = h_k(r_k = t|r_{k+1} = s)$. On the other hand, $\lim_{\delta \to 0} P(B_s|A_t)/\delta$ is simply the density function of r_{k+1} given that r_k is fixed at $r_k = t$. By Lemma 1, this density function is given by $\frac{s}{2t}$ for $s \in [t - 1, t + 1]$ if $t \ge 1$ and for $s \in [1 - t, 1 + t]$ if $0 < t < 1$. Also, $\lim_{\delta \to 0} P(A_t)/\delta = g_k(t)$ and $\lim_{\delta \to 0} P(B_s)/\delta = g_{k+1}(s)$. Thus

$$\begin{aligned} h_k(r_k = t|r_{k+1} = s) &= \lim_{\delta \to 0} P(A_t|B_s)/\delta \\ &= \lim_{\delta \to 0} (P(B_s|A_t)/\delta)(\frac{P(A_t)/\delta}{P(B_s)/\delta}) \\ &= \frac{s}{2t} \frac{g_k(t)}{g_{k+1}(s)}, \end{aligned}$$

where $t \in [s - 1, s + 1]$ if $s \ge 1$ and $t \in [1 - s, 1 + s]$ if $s < 1$. Substituting r_k for t and r_{k+1} for s now yields the desired result.

Intuitively, it is easy to believe that the longer the random walk the more the walk forgets that it originated at the origin. The following theorem provides the theoretical

base for the above intuitive assertion. It basically states that when $r_{k+1} = \tau$ is small compared to k, then $h(r_k|r_{k+1} = \tau)$ can be approximated by $\frac{r_k}{2\tau}$ where $r_{k+1} = \tau$, and $\frac{r_k}{2\tau}$ is the corresponding probability density function of Lemma 1 without the condition that X_k is a vertex on a random walk that leads to the origin.

Theorem 2 *Using the previously defined notation $h_k(r_k|r_{k+1} = \tau)$ is equal to $\frac{r_k}{2\tau}(1 + O(\tau_0^2/k))$ for $r_k \in [|\tau - 1|, \tau + 1]$ (and it equals 0 otherwise), where $\tau_0 = \max\{\tau, 1\}$.*

The proof of the theorem relies on the following lemma, which is a refinement of the original approximation given in [14].

Lemma 2 [4] *For $k \geq 10$, the following inequality holds:*

$$\left| f_k(X_k) - \left(\sqrt{\frac{3}{2\pi k}}\right)^3 \exp\left(-\frac{3\,|X_k|^2}{2k}\right)\right| < \frac{0.5}{k^{\frac{5}{2}}}. \tag{4}$$

In other words, $f_k(X_k) = \left(\sqrt{\frac{3}{2\pi k}}\right)^3 \exp\left(-\frac{3|X_k|^2}{2k}\right) + O(\frac{1}{k^{\frac{5}{2}}})$.

Proof of Theorem 2. In the case of equilateral random polygons, the density function of X_k in EP_n is given by

$$h_k(X_k) = \frac{1}{f_n(O)} \cdot f_k(X_k) \cdot f_{n-k}(X_k) \tag{5}$$

which can be obtained by integrating the joint density function of the vertices of EP_n

$$f_{EP}(X_1, X_2, ..., X_{n-1}) = \frac{1}{f_n(X_n)}\varphi(X_1)\varphi(X_2 - X_1)\cdots\varphi(X_n - X_{n-1})$$

(with $X_n = X_0 = O$) over X_1, X_2, ..., X_{n-1} except for X_k.

Similarly, let X_k and X_{k+1} be two consecutive vertices of an equilateral random polygon EP_n, then the joint probability density function $f_{EP}(X_k, X_{k+1})$ of X_k and X_{k+1} is given by

$$\int\cdots\int \frac{\varphi(X_1)\varphi(X_2 - X_1)\cdots\varphi(X_n - X_{n-1})}{f_n(O)}\widehat{\mathrm{d}X_k}\widehat{\mathrm{d}X_{k+1}}, \tag{6}$$

where the integral is taken over all variables except X_k and X_{k+1}. This results in

$$f_{EP}(X_k, X_{k+1}) = \varphi(X_{k+1} - X_k)f_k(X_k)f_{n-k-1}(X_{k+1})/f_n(O). \tag{7}$$

The conditional density function $h(X_k|X_{k+1})$ of X_k under the condition that X_{k+1} is fixed (with $|X_{k+1}| = \tau$) is then given by

$$\begin{aligned} h(X_k|X_{k+1}) &= \frac{f_{EP}(X_k, X_{k+1})}{f_{EP}(X_{k+1})} \\ &= \frac{\varphi(X_{k+1} - X_k)f_k(X_k)f_{n-k-1}(X_{k+1})/f_n(O)}{f_{k+1}(X_{k+1}) \cdot f_{n-k-1}(X_{k+1})/f_n(O)} \\ &= \frac{\varphi(X_{k+1} - X_k)f_k(X_k)}{f_{k+1}(X_{k+1})}. \end{aligned}$$

In our case, $|X_k|$ and $|X_{k+1}|$ are of the order of $\tau_0 = \max\{\tau, 1\}$, which is small relative to $k+1$. Thus we have $\exp\left(-\frac{3|X_k|^2}{2k}\right) = 1 + O(\tau_0^2/k)$, where we mean that for small values of the fraction $\frac{3|X_k|^2}{2k}$ the difference $1 - \exp\left(-\frac{3|X_k|^2}{2k}\right)$ has exactly the order of τ_0^2/k. Similarly we write that $\exp\left(-\frac{3|X_{k+1}|^2}{2(k+1)}\right) = 1 + O(\tau_0^2/k)$. It follows from Lemma 2 that

$$f_k(X_k) = \left(\sqrt{\frac{3}{2\pi k}}\right)^3 (1 + O(\tau_0^2/k))$$

and

$$\begin{aligned} f_{k+1}(X_{k+1}) &= \left(\sqrt{\frac{3}{2\pi(k+1)}}\right)^3 (1 + O(\tau_0^2/k)) \\ &= \left(\sqrt{\frac{3}{2\pi k}}\right)^3 (1 + O(\tau_0^2/k)), \end{aligned}$$

hence

$$\begin{aligned} h(X_k|X_{k+1}) &= \frac{\varphi(X_{k+1} - X_k) f_k(X_k)}{f_{k+1}(X_{k+1})} \\ &= \varphi(X_{k+1} - X_k)(1 + O(\tau_0^2/k)). \end{aligned}$$

The result now follows by the same argument used to prove Lemma 1 since $g(r_k|r_{k+1} = \tau)$ is derived using $h(X_k|X_{k+1}) = \varphi(X_{k+1} - X_k)(1 + O(\tau_0^2/k))$ as the density function with X_{k+1} fixed at $|X_{k+1}| = \tau$.

3. An algorithm for generating the equilateral random polygon with its theoretical probability distribution

Given the probability distribution function of $r_k = |X_k|$ (for all k) as derived in the last section, one can in theory generate an equilateral random polygon of any given length $n \geq 3$ as described in the following algorithm. The distributions of the equilateral random polygons so generated follow exactly the theoretical probability distributions.

Initial step: The starting and ending point of the polygon is set to be the origin by default. X_{n-1} is chosen uniformly on the unit sphere centered at the origin.

Recursive steps: Starting with $j = 2$, choose r_{n-j} according to the distribution derived from Theorem 1 (with the condition that r_{n-j+1} has been chosen in the prior step). Specifically, if $F(r_{n-j})$ is the cumulative probability distribution of r_{n-j} (under the condition that r_{n-j+1} is fixed), then r_{n-j} is chosen to be the solution of the equation $F(r_{n-j}) = u$, where u is a random number uniformly chosen from $[0, 1]$. Once $r_{n-j} = |X_{n-j}|$ is chosen, X_{n-j} is chosen uniformly on the intersection circle of the unit sphere centered at X_{n-j+1} and the sphere centered at O with radius r_{n-j}. The last value for j is $n - 2$.

Final step: X_2 is already chosen. At this point X_1 is simply chosen uniformly from the intersection circle of the unit sphere centered at X_2 and the unit sphere centered at O. This allows the random walk to return to the origin.

3.1. The applicability of the algorithm: general case

The direct application of the algorithm outlined in the last section, unfortunately, is not so easy. Although the density functions $g_j(r_j) = 4\pi r_j^2 f_j(X_j)$ have an explicit formula as given in (3), the formula involves an improper integral. For any value of n, we can in principle compute the integral exactly using a software package like Mathematica. Thus we have an exact expression for the functions $h_k(r_k|r_{k+1})$ in Theorem 1. However the length of these expressions increases very quickly and their size and the computation time involved makes it difficult to increase the size of k arbitrarily. (For example the simplified explicit formula for $g_k(r_k)$ for $k = 120$ (as in Theorem 1) fills 9 pages of a printout of a Mathematica notebook single spaced with size 10 fonts.) Furthermore, $h_k(r_k|r_{k+1})$ involves the quotient of two such functions and additionally we need the cumulative density functions $H_k(r_k|r_{k+1})$ arising from the $h_k(r_k|r_{k+1})$ functions. Thus we can only compute the cumulative density functions $H_k(r_k|r_{k+1})$ for small values of k with very high accuracy. Fortunately, for large values of k a numerical approximation of the cumulative density functions $h_k(r_k|r_{k+1})$ is also possible using Lemma 2. Namely for large values of k we replace the functions $f_k(X_k)$ with the much simpler exponential function given there. Using the exponential approximation we can compute an approximation of the density functions $h_k(r_k|r_{k+1})$ and the cumulative density functions $H_k(r_k|r_{k+1})$ for arbitrarily large values of k. The details of our numerical methods to compute the cumulative density functions $H_k(r_k|r_{k+1})$ are explained in Section 4.

3.2. The applicability of the algorithm: confined case

Our algorithm can be easily modified into an algorithm that generates equilateral random polygons confined in a sphere centered at the origin.

Let us assume that the confining sphere has a fixed radius $R \geq 1$ (in general R is much smaller in comparison with the length n of the polygons generated). Basically, the algorithm selects the vertices in a consecutive manner in the same order as described earlier. More precisely, assume that X_{k+1} has been chosen, then X_k is chosen precisely by the same procedure outlined above in the case that $r_{k+1} \leq R - 1$, since in this case the confining condition does not apply to X_k. If $r_{k+1} > R-1$ then we are in a situation as shown in Figure 2 on the top. The current position is at X_{k+1} which is distance $\tau = r_{k+1}$ away from the origin (see top right of the figure). Our next step generates a point on the sphere with center X_{k+1} and radius one, but the point must be inside the confining sphere. Thus, we modify the procedure as follows. Let $H_k(r_k|r_{k+1})$ be the cumulative distribution function of r_k under the condition that r_{k+1} is fixed. Recall

that its corresponding probability density function $h_k(r_k|r_{k+1})$ is defined in Theorem 1(b). The maximal value possible for r_k is R and the probability that r_k is at most R is thus given by $H_k(R|r_{k+1})$. It follows that the conditional cumulative probability distribution function for r_k is $H_k(r_k|r_{k+1})/H_k(R|r_{k+1})$. As before, r_k is then chosen to be the solution of the equation $u = H_k(r_k|r_{k+1})/H_k(R|r_{k+1})$, where u is uniformly chosen from $[0, 1]$. Once r_k is chosen, X_k is on the intersection circle of the sphere centered at the origin with radius r_k and the unit sphere centered at X_{k+1}. This process (together its numerical implementation) is illustrated in Figure 2. The value u' shown in the figure is randomly selected from $[0, H_{k,\tau}(R)]$, which is equivalent to $uH_k(R|r_{k+1})$ in the equation $u = H_k(r_k|r_{k+1})/H_k(R|r_{k+1})$.

4. Numerical methods

Let $\tau = r_{k+1} = |X_{k+1}|$ and $H_{k,\tau}(r_k) = H_k(r_k|r_{k+1} = \tau)$. Computing $H_{k,\tau}$ every time the next r_k must be determined using the last computed point X_{k+1} is computationally expensive. Thus the process of generating random polygons is divided into two steps. First a suitable set of functions $H_{k,\tau}$ is computed and saved. This step is computationally quite demanding but is performed only once. The second step then uses the pre-computed functions $H_{k,\tau}$ to generate the polygons in confinement. This section provides additional details about both steps. Note that there is no reason to compute $H_{k,0}$ or $H_{k,k+1}$, since in both cases there is only one possibility for r_k: $r_k = 1$ for $\tau = 0$ and $r_k = k$ for $\tau = k + 1$. Thus in the following we only deal with τ values in $(0, k + 1)$.

Step 1. Precomputing the $H_{k,\tau}$:

The method to generate random uniform polygons developed here relies on a numerical determination of the cumulative conditional probability density functions $H_{k,\tau}$ for values of $\tau = r_{k+1} \leq R$ (where R is the radius of the confining sphere) for $k \in \{2, 3, \ldots, k_{max}\}$ for a suitably chosen positive integer k_{max}. Choose two small values $\Delta\tau$ and Δr and let $k = 2, \ldots, k_{max}$ and let $\tau = \Delta\tau s$ for $s = 1, \ldots, \lceil \frac{R}{\Delta\tau} \rceil$. For each pair (k, τ) we sample data points of the cumulative probability density function $H_{k,\tau}$ and by switching the in- and outputs also obtain its inverse $H^{-1}_{k,\tau}$. Additionally we pick a small positive value $\epsilon < \Delta\tau$ (e.g. $\epsilon = 10^{-4}$) and sample $H_{k,\epsilon}$ and if $k + 1 - \epsilon \leq R$ we also sample $H_{k,k+1-\epsilon}$.

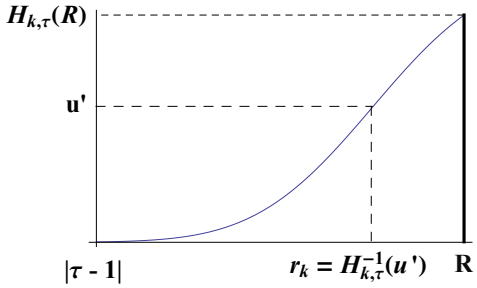
Let $I_{k,\tau} = [a, b]$ be the interval domain of the random variable r_k where $H_{k,\tau}$ is non trivial. (That is, if r_k is less than a then $H_{k,\tau} = 0$ and if r_k is larger than b then $H_{k,\tau} = 1$). If τ is a value that is at least one unit smaller than k and more than one unit away from zero then simply $I_{k,\tau} = [\tau - 1, \tau + 1]$. However if $\tau + 1 > k$ or $\tau < 1$ then the interval $I_{k,\tau}$ becomes smaller because $|1-\tau| \leq r_k \leq k$ must be true. Let $w_{k,\tau}$ be half the width of $I_{k,\tau}$. Then $w_{k,\tau} = (\min\{\tau+1, k\} - |\tau - 1|)/2$ and $I_{k,\tau} = [|\tau - 1|, |\tau - 1| + 2\ w_{k,\tau}]$.

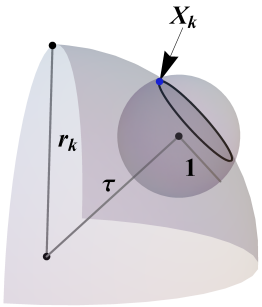
The range of $H_{k,\tau}$ over $I_{k,\tau}$ is the interval $[0, 1]$ regardless of how small $I_{k,\tau}$ becomes. We use the chosen $\Delta\ r$ and for a fixed k and τ sample $H_{k,\tau}$ for $r = |r_{k+1}-1|+i\Delta r w_{k,\tau}$ for

Finding X_k given X_{k+1}

(Notation: R=confinement radius, $H_{k,\tau}(r)$ is the cdf of $|X_k|$ given $|X_{k+1}| = \tau$)

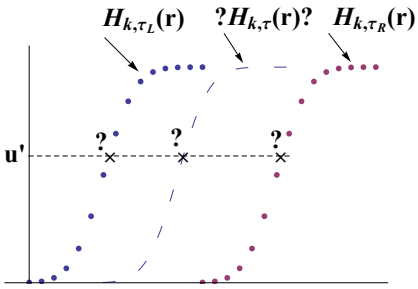
Exact Approach

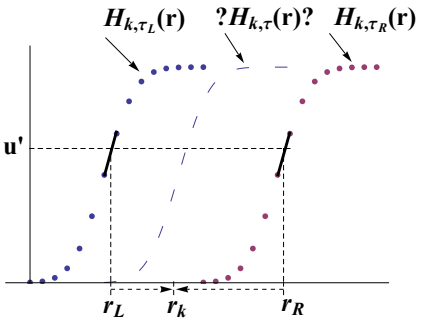




1) Uniformly select $u' \in [0, H_{k,\tau}(R)]$; 2) Calculate $r_k = |X_k|$ using $H^{-1}_{k,\tau}(u')$ as on the left; 3) Pick a random point on the circle of intersection between a sphere of radius r_k at the origin and a unit sphere at X_{k+1} as on the right

Numerical Implementation





For selected τ values $H_{k,\tau}(r)$ are precomputed as described in the text. This now complicates the inverse lookup method for obtaining X_k. We use the neighboring precomputed H functions to approximate $H_{k,\tau}$. For each of these neighbors, H_{k,τ_L} and H_{k,τ_R}, we use linear interpolation to approximate $r_L = H^{-1}_{k,\tau_L}(u')$ and $r_R = H^{-1}_{k,\tau_R}(u')$ (shown at right), and described in the text with the $H^{*-1}_{k,\tau}$ function. After a transformation to equalize the domains, r_L and r_R are linearly interpolated to yield an approximate r_k as described in the text with $\tilde{H}_{k,\tau}$.

Figure 2. A graphic illustration of the polygon generating algorithm and its numerical implementation.

$0 \leq i \leq \lfloor \frac{2}{\Delta r} \rfloor$ and if $\lceil \frac{2}{\Delta r} \rceil \neq \lfloor \frac{2}{\Delta r} \rfloor$ add the point $(|\tau - 1| + 2w_{k,\tau}, 1)$ to the sampled points of $H_{k,\tau}$. This ensures that we sample the same number of points for each $H_{k,\tau}$, regardless of the size of the $I_{k,\tau}$. For each of the required cumulative conditional probability density functions we have $\lceil \frac{2}{\Delta r} \rceil + 1$ data points, and for each value of k, there are $\lceil \frac{min[R,k+1]}{\Delta \tau} \rceil$ such discrete probability density functions.

Step 2: Using $H_{k,\tau}$ to determine r_k for a given $|X_{k+1}|$: The discretization of the $H_{k,\tau}$ as described above requires that we address how to obtain $H_{k,\tau}$ values for τ and r values which were not sampled. For $\tau < \epsilon$ and for $\tau > k + 1 - \epsilon$ we approximate $H_{k,\tau}$ by a straight line over $I_{k,\tau} = [a, b]$ connecting $(a, 0)$ with $(b, 1)$. Thus the remainder of the discussion about approximations only includes τ values between ϵ and Min$\{k + 1 - \epsilon$, $R\}$.

The general approach for these approximations selects the two closest sample points available called r_R and r_L, where R and L stand for *right* and *left*. Using r_R and r_L we compute a weighted average. For a graphical interpretation of these steps see the bottom of Figure 2.

To approximate $H_{k,\tau}$ for an r value we use two values $r_R = min\{\lceil \frac{r}{\Delta r w_{k,\tau}} \rceil \Delta r w_{k,\tau}, |\tau - 1| + 2w_{k,\tau}\}$ and $r_L = \lfloor \frac{r}{\Delta r w_{k,\tau}} \rfloor \Delta r w_{k,\tau}$ such that $r_L \leq r \leq r_R$ and $u_R = H_{k,\tau}(r_R)$ and $u_L = H_{k,\tau}(r_L)$. (Note that in the computation of r_R the expression $\lceil \frac{r}{\Delta r w_{k,\tau}} \rceil \Delta r w_{k,\tau}$ may be outside of the sampled range and in such a case, by taking the minimum value, we select the data point $(|\tau - 1| + 2w_{k,\tau}, 1)$ as explained in the last paragraph of Step 1.)

We are now ready to specify the approximation $H^*_{k,\tau}$ of $H_{k,\tau}$ for r values which were not sampled, assuming that τ is a value for which $H_{k,\tau}$ was sampled.

$$H^*_{k,\tau}(r) = \begin{cases} H_{k,\tau}(r) & \text{if } r_L = r = r_R; \\ \frac{u_R - u_L}{r_R - r_L}(r - r_L) + u_L & \text{otherwise.} \end{cases}$$

For the approximation for a u value for $H^{-1}_{k,\tau}$ we use the sampled values u_R and u_L such that $u_L \leq u \leq u_R$ and $r_R = H^{-1}_{k,\tau}(u_R)$ and $r_L = H^{-1}_{k,\tau}(u_L)$ and $r_R \leq r_L + \Delta \tau w_{k,\tau}$.

$$H^{*-1}_{k,\tau}(u) = \begin{cases} H^{-1}_{k,\tau}(u) & \text{if } u_L = u = u_R; \\ \frac{r_R - r_L}{u_R - u_L}(u - u_L) + r_L & \text{otherwise.} \end{cases}$$

This allows us to determine $H^*_{k,\tau}$ for all values of r in $I_{k,\tau}$ and $H^{*-1}_{k,\tau}$ for all values of u in $[0, 1]$.

Next we deal with the discretization of τ. The approximation for a τ value uses the two sampled distributions H_{k,τ_L} and H_{k,τ_R}, where $\tau_R = min\{\lceil \frac{\tau}{\Delta \tau} \rceil \Delta \tau, k + 1 - \epsilon\}$ and $\tau_L = max\{\lfloor \frac{\tau}{\Delta \tau} \rfloor \Delta \tau, \epsilon\}$, such that $\tau_L \leq \tau \leq \tau_R$. Since the $I_{k,\tau}$ are of different sizes we adjust H_{k,τ_L} and H_{k,τ_R} such that their adjusted domains are equivalent to $I_{k,\tau}$. This is done by shifting and scaling the r values using

$$r' = shsc(r, k, \tau_{from}, \tau_{to}) = \frac{w_{k,\tau_{to}}}{w_{k,\tau_{from}}}(r - c_{k,\tau_{from}}) + c_{k,\tau_{to}},$$

where $c_{k,\tau} = |\tau - 1| + w_{k,\tau}$. $H_{k,\tau_{from}}$ is the cumulative density function which is shifted to align with the domain $I_{k,\tau_{to}}$.

Below we specify the approximation $\tilde{H}_{k,\tau}$ of $H_{k,\tau}$ for any τ value and any r value in $[|\tau - 1|, min\{k, \tau + 1\}]$.

$$\tilde{H}_{k,\tau}(r) = \begin{cases} H^*_{k,\tau}(r) & \text{if } \tau_L = \tau = \tau_R; \\ \frac{u'_R - u'_L}{\tau_R - \tau_L}(\tau - \tau_L) + u'_L & \text{otherwise} \end{cases}$$

where $u'_L = H^*_{k,\tau_L}(r'_L)$ with $r'_L = shsc(r, k, \tau, \tau_L)$ and $\tau = r_{k+1}$ and similarly $u'_R = H^*_{k,\tau_R}(r'_R)$ with $r'_R = shsc(r, k, \tau, \tau_R)$.

Similarly we specify the approximation $\tilde{H}^{-1}_{k,\tau}$ of $H^{-1}_{k,\tau}$ for any τ value and any u value in $[0, 1]$.

$$\tilde{H}^{-1}_{k,\tau}(u) = \begin{cases} H^{*-1}_{k,\tau}(u) & \text{if } \tau_L = \tau = \tau_R; \\ \frac{r'_R - r'_L}{\tau_R - \tau_L}(\tau - \tau_L) + r'_L & \text{otherwise} \end{cases}$$

where $r'_L = shsc(r_L, k, \tau_L, \tau)$ with $r_L = H^{*-1}_{k,\tau_L}(u)$ and similarly $r'_R = shsc(r_R, k, \tau_R, \tau)$ with $r_R = H^{*-1}_{k,\tau_R}(u)$.

Now we are ready to describe the steps to randomly chose r_k given that the current distance to the origin is $|X_{k+1}| = r_{k+1} = \tau$. A value u, $0 \leq u \leq 1$ is chosen with uniform probability. For the unconstrained case when $|X_{k+1}| + 1 \leq R$, r_k is equal to $\tilde{H}^{-1}_{k,\tau}(u)$. For the constrained case, when $\tau + 1 > R$, we determine $u^{mod} = \tilde{H}_{k,\tau}(R)$ and set $r_k = \tilde{H}^{-1}_{k,\tau}(uu^{mod})$. The bottom of Figure 2 shows the two averaging steps involved in determining r_k from $u' = uu^{mod}$, where $u^{mod} = 1$ for the unconstrained case. The implementation uses a binary search of the sampled values for the cumulative distribution function for the reverse look-up.

In our computations for the data collection we used $k_{max} = 200$, $\Delta\tau = 1/10$, $\Delta r = 1/1000$ and $\tau \leq 7$. ($R = 6$ was the largest radius of the confining sphere used in our experiments.) The value of $k_{max} = 200$ is large enough to create polygons needed to test our algorithm and for testing conjectures about the geometry and topology of such polygons. For large values of k the polygons become highly knotted and resolving the knot type seems beyond current computational techniques. For all pairs of k and τ within this range we computed the functions $H_{k,\tau}$ exactly using the software package Mathematica. We then evaluated the functions to a precision of 16 decimal digits for the r values starting at $|\tau - 1|$ and incremented by $\Delta r = (1/1000)w_{k,\tau}$. The largest k for which $H_{k,\tau}$ was sampled is $k = 235$. The sizes of the exact cumulative density functions grow quite fast. The Mathematica output file (in a .m package format) that contains all the exact functions $H_{k,\tau}$ needed for $k_{max} = 235$ had a size of about 2 Gigabytes. Using the above techniques and better file management is seems possible to push the value of k_{max} up to about 250 but probably not a lot further.

However, in principle we can extend the computations beyond a value of $k_{max} = 250$. For triples of $k > k_{max}$, τ, and r we compute the values of $H_{k,\tau}(r_k = r)$ quite accurately

using the approximation given by Lemma 2 instead of the exact integral given in Equation 3. This computation can be carried out for very large values of k since no complicated integral needs to be computed. For example, using the bound of Lemma 2 and $k = 250$ results in an error of approximately 5×10^{-7} in the approximations of the functions $f_k(X_k)$. This error is no larger than the error introduced through the averaging procedure described in this section which is investigated in the numerical results section. For the data collection used for the results in this paper $k_{max} = 200$ was sufficient.

5. Numerical results and discussions

One of the advantages of this method is that it is very fast once the precomputing of the $H_{k,\tau}$ has been completed. Polygon generation can be carried out very quickly, for example the generation of a single polygon with a length of $k = 200$ took less than half a second of computing time using a normal lap- or desktop machine using the software Mathematica. The runtime growth for a polygon generation is linear with the length k however the length k is limited by the largest values for which the precomputing of the $H_{k,\tau}$ has been completed, see Figure 3. In the numeric study described we restricted ourselves to polygons of length less than 200 for the simple reason that this range is enough to allow comparisons with other studies of polygons in confinement.

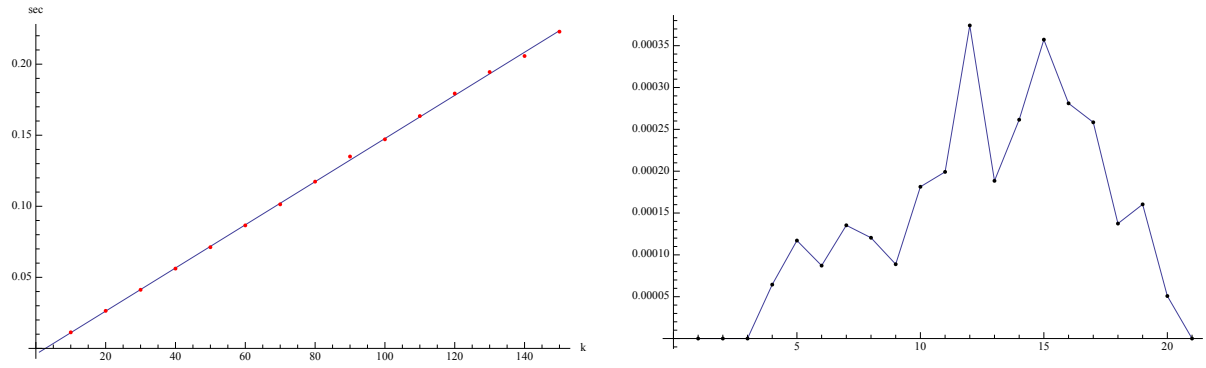


Figure 3. On the left the average runtime of polygon generation. On the right the distance of position between the exact polygon and the polygon generated using the procedure described here, see also Figure 9.

Using the algorithm described in the previous section we generated $1,000$ polygons for each of the following values of the radius of confinement R and the length of the polygon n: $R = 1, 1.5, 2, 3, 4, 5, 6$ and $n = 10, 20, 30, 50, 70, 90, 110, 140, 170, 199$. For some of the sets of $1,000$ polygons we computed as a control value the average writhe and observed that it is close to zero as one would expect. For each of these collections we then computed the average crossing number (ACN). (Both of these calculations were done using the Gaussian integral formula.) In [1] it is shown that the average crossing number of a random polygon P_n in confinement with n edges grows as an

$O(n^2)$ and the authors use for the average crossing number a best fit-function of the form $y = a(R)n^2 + b(R)n \log n$, where $a(R)$ and $b(R)$ are constants that depend on the radius of confinement R. Using the same fit function, we obtained excellent agreement with our data as shown in Figure 4. The R^2-values for the fits range from 0.999935 to 0.9999991. The data points for $R = 1$ and $n > 140$ are omitted because the large values of the average crossing number make the rest of the functions difficult to see (for example for $n = 199$ and $R = 1$ we obtained a mean ACN of 4173.23) and we note that the fit for the omitted values for $R = 1$ is as good as in the rest of the data.

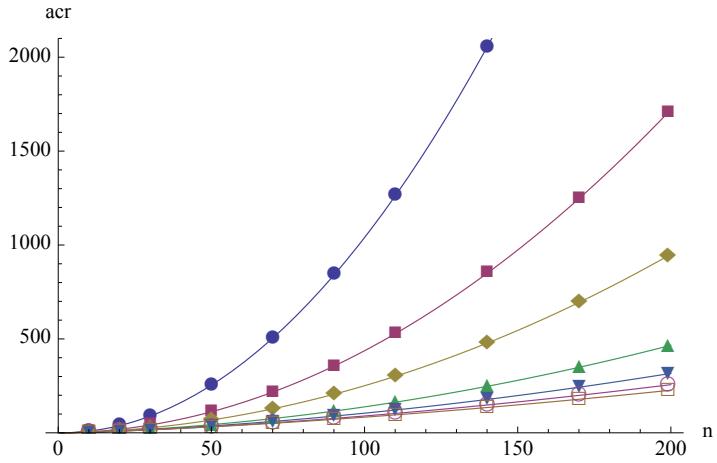


Figure 4. Each data points represents the average of 1,000 randomly generated polygons. The largest function represents $R = 1$, the second largest $R = 1.5$ and so on.

Figure 5 shows the values of $a(R)$ and $b(R)$ together with best fit functions. The best fit function for the $a(R)$ data has a R^2-value of 0.999981 and we have $a(R) \approx \frac{.1074}{R^{2.2955}}$ where the coefficients are rounded to the nearest 1/10000. Similarly, the best fit function for the $b(R)$ data has a R^2-value of 0.992703 and we have $b(R) \approx 0.140 - 0.5561e^{-R}$ where the coefficients are rounded to the nearest 1/10000.

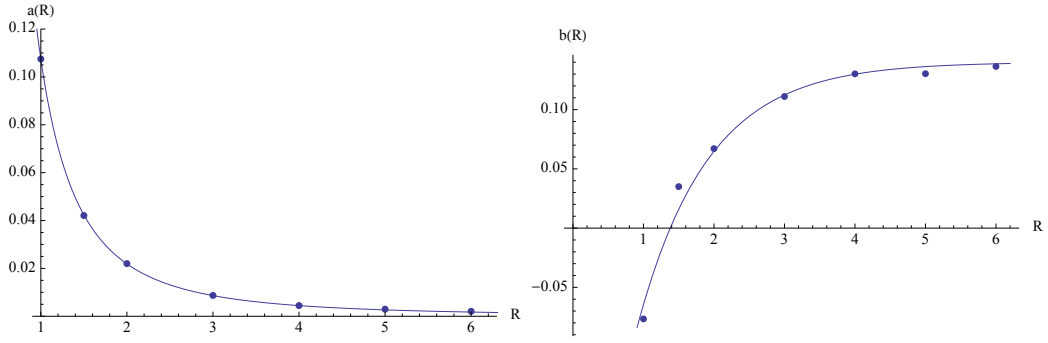


Figure 5. Shown are the coefficients $a(R)$ and $b(R)$ together with their best-fit functions. The fit on the $a(R)$ terms is excellent while the fit on the $b(R)$ terms is not quite as good.

One of the most difficult questions to answer is how close the distribution of our sampled polygons is to a distribution of polygons that are sampled with uniform probability. In our method we approximate the cumulative density functions $H_{k,\tau}(r_k)$ using the two closest sample points available (for τ) and computing a weighted average. Unfortunately this introduces an error and we are unable to analytically find an upper bound for this error. However, we can illustrate the size of this error with an example. We consider the following: Assume that we generate polygons without confinement and that $k = 20$ is fixed. For a fixed $\tau = r_{k+1}$ the graph of a function $H_{k,\tau}(r_k)$ is non-trivial only if $|r_k - \tau| \leq 1$, or more precisely if $r_k \in I_{k,\tau} = [|\tau - 1|, min\{\tau + 1, k\}]$. We can think of $I_{k,\tau}$ as the domain of the function $H_{k,\tau}(r_k)$. As discussed in Section 4, for $1 \leq \tau \leq k - 1$, the width of $I_{k,\tau}$, that is, the possible range for r_k values, is two. Thus for $k = 20$ fixed, the union over all graphs of $H_{k,\tau}(r_k)$ functions with $\tau \leq 21$ can be viewed as a surface in a rectangular box with dimensions $21 \times 20 \times 1$ where the unions of the domains of the $H_{k,\tau}(r_k)$ functions forms a strip (mostly of width two) that runs diagonally across the bottom 21 by 20 rectangle of this box. Of course for values $\tau = r_{k+1} < 1$ the width of the strip narrows due to the fact that $r_k \in [1 - \tau, 1 + \tau]$. Similarly, for values $\tau = r_{k+1} > 19$ the width of the strip narrows due to the fact that $r_k \in [\tau - 1, 20]$. Schematically this is shown in Figure 6.

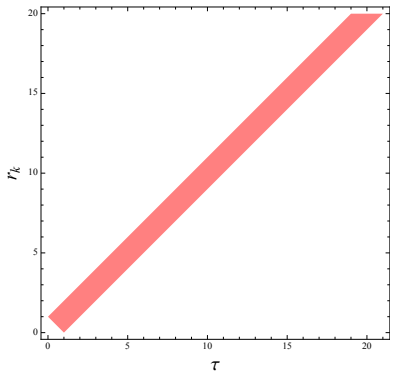


Figure 6. The strip in the (τ, r_k)-plane obtained as the union of the domains where the cumulative density functions $H_{k,\tau}(r_k)$ are non trivial.

In order to estimate the error induced by the averaging procedure when selecting one r_k value we use the following method: We create two data sets: Using $\Delta\tau = 1/10$ we create the data set S_1 of the cumulative density functions $H_{k,\tau}(r_k)$ using Theorem 1(b). Similarly, using $\Delta\tau = 1/40$ we create an even larger data set S_2 of the cumulative density functions $H_{k,\tau}(r_k)$. We now use the averaging procedure and the data set S_1 to create the approximate cumulative density functions $H_{k,\tau}(r_k)$ that are contained in S_2 but missing from S_1. That is in between two values of τ in S_1, say $\tau = s/10$ and $\tau = (s + 1)/10$ with $s \in \mathbb{N} \cap [1, 208]$ we create three approximate cumulative density functions $H_{k,\tau}(r_k)$ for values of $\tau = s/10 + 1/40, s/10 + 1/20$ and $s/10 + 3/40$. Let S_3 be the data set that is the union of S_1 and these new approximate cumulative density functions. Now the two data sets S_3 and S_2 both contain the same number of cumulative

density functions $H_{k,\tau}(r_k)$ with a spacing of $\tau = 1/40$, where the functions in S_2 are all exact while 3/4th of the functions in S_3 are approximations created by using the averaging procedure. After taking the difference between these two data sets we create an interpolation surface S where the difference (the error) is shown on the z-axis. The surface S projects exactly onto the strip in the (τ, r_k)-plane as shown in Figure 6. If one would show the entire surface S then because of the scale on the τ- and r_k-axes one could not make out any details. Therefore we elected to show only parts of this surface. Figure 7 shows the initial part of the surface S ranging from $\tau \in [1/10, 2]$ while Figure 8 shows the part ranging from $\tau \in [6, 8]$.

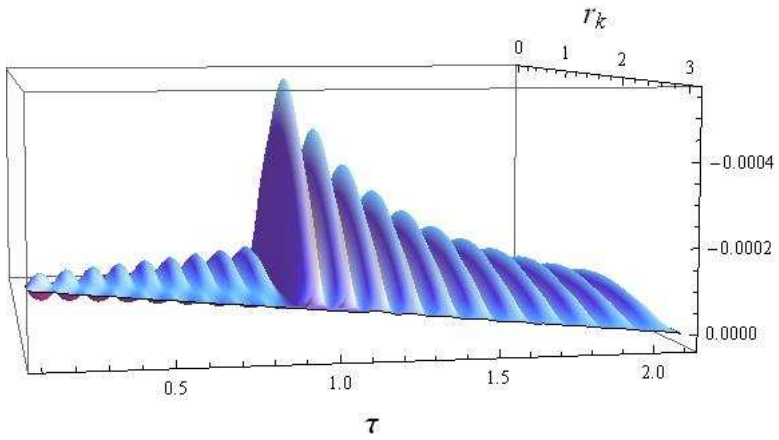


Figure 7. The surface S with $\tau = r_{k+1} \in [1/10, 2]$.

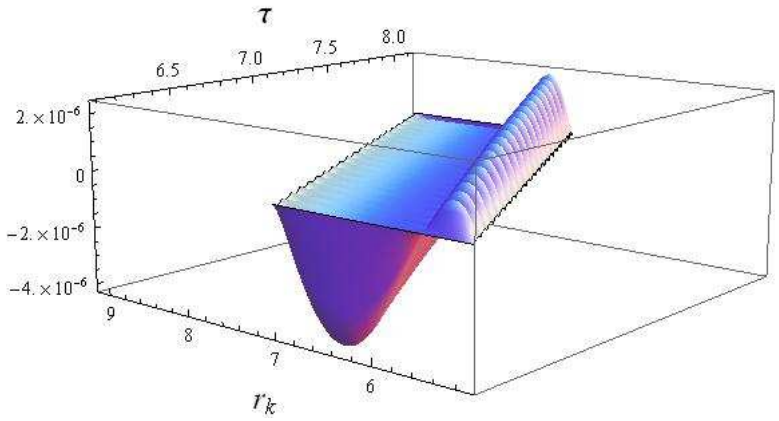


Figure 8. The surface S with $\tau = r_{k+1} \in [6, 8]$.

The ripples in the surface are typical and are caused by the fact that whenever we use the exact cumulative density functions $H_{k,\tau}(r_k)$ in increments of $\Delta\tau = 1/10$ the error is zero. The initial strip contains the largest error that occurs over the whole surface S and this maximal error is about 0.0005. In the middle of the surface S the parabolic shape of the error changes more to an s-shape. In the example shown in Figures 7 and

8 for each cumulative density functions $H_{k,\tau}(r_k)$ we resolve the surface with $\Delta r_k = 0.01$ and as previously mentioned $\Delta\tau = 1/40$. We note that finer resolutions (using even smaller values of Δr_k) have no visible effect on the surface at the scale used in Figures 7 and 8. By looking at some other values of k we claim that this example of an error surface is typical. Clearly one can reduce this error by decreasing the value of $\Delta\tau$. Doing this however increases the number of functions that need to be precomputed. Our choice of $\Delta\tau = 1/10$ represents a compromise between the goals to keep the error small and to push the values of k up, while keeping the amount of precomputed data manageable. It is an open question if decreasing this error further has an effect on the sampled polygons that can actually be measured by some geometric quantity.

To estimate the error of the overall procedure we have computed a twenty step polygon in a confinement sphere of radius three in two different ways, see Figure 9. Choosing exactly the same value of u for each vertex and the same angle to choose the actual vertex X_k on the circle of potential positions, we computed a polygon using two different approaches. One approach used the averaging procedure described here and the other used an exact integral computation (which only introduces a rounding error that is the default machine precision of about 10^{-16}). In Figure 3 on the right we show the actual distances between corresponding vertices which are quite small. As can be seen, the errors made when selecting the indiviudal r_k for each step do not add up, but tend to cancel each other. While this is only one relatively short polygon (due to the difficulty of the exact calculation), it provides further evidence that our procedure yields polygons that are very close to the actual polygons that are computed with the correct probability distribution with no rounding errors.

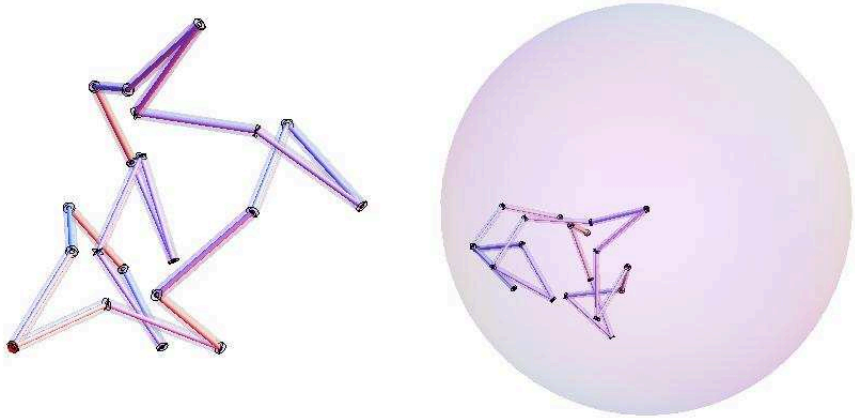


Figure 9. On the left we show two twenty-segment polygons that are basically identical in the confinement sphere of radius three. On the right, the same two polygons with the confinement sphere. The exact computation is shown as a polygon with a tube radius of 1/50 while the approximate polygon is shown with a tube radius of 1/20. We can clearly see that the bigger radius tubes contain the smaller radius tubes.

6. Ending remarks

We end this paper with several remarks concerning some possible future studies that may stem from this paper.

Remark 1 A more realistic model for confined DNA, of course, has to consider restricting factors such as the volume exclusion effect, the bending angle and torsional rigidity restrictions. In some cases such as the bending angle and torsional rigidity restrictions, different weights may be placed on the probability distribution functions in our algorithm to achieve the desired restriction conditions, at least in the average sense. Such an approach would be fairly easy to implement. However, in some other cases such as the volume exclusion effect case, the above approach does not apply. Of course, if one considers the sampling space for such a restricted random polygon being a subset of the configuration space of all confined equilateral polygons satisfying the restriction condition, it is conceivable that an acceptance/rejection approach can be used based on our algorithm here. However, considerable work is needed to get around the computational run time issue, as it is known that the acceptance/rejection method is extremely time consuming. This shall be one area the authors will look into in their future study.

Remark 2 In the non-confined case, our generation method probably does not have much advantage over the existing methods such as the crankshaft and the hedgehog methods since these methods are quite effective. We also suspect that no significant difference can be detected if one compares the various (geometric or topological) characteristics of polygons generated using our method and other methods in the case of confinement, as our limited numerical study on the mean ACN indicates. However, to reach a more reliable conclusion, comparisons using topological characteristics such as knot spectrum are needed. Such a study is computation intensive in nature and exceeds the scope of this paper. The authors do intend to carry out such a study in the future.

Remark 3 In this paper, we have set the starting point of the equilateral polygon at the center of the confining sphere. This is purely for the sake of convenience. Although some technical details have to be addressed under a more general setting when the starting point does not coincide with the center of the confining sphere, the basic idea in our approach still applies and we do not foresee that this poses as a big problem. A more interesting question here is, if we choose a different starting point for the polygon within the confining sphere, what kind of effect (if any) does this have on the overall geometric and topological characteristics? This is also a direction for future study.

Acknowledgments

This work is supported in part by NSF Grants #DMS-0920880 and #DMS-1016460 (Y Diao), and by NSF grant #DMS-1016420 (C. Ernst, A. Montemayor and U. Ziegler).

[1] Arsuaga J, Borgo B, Diao Y and Scharein R 2009, J. Phys. A: Math. Theor **42** 465202.
[2] Arsuaga J, Vazquez M, McGuirk P, Sumners D W and Roca J 2005, *Proc Natl Acad Sci USA* **102** 9165–9.
[3] Arsuaga J, Vazquez M, Trigueros S, Sumners D W and Roca J 2002, *Proc Natl Acad Sci USA* **99** 5373–7.
[4] Diao Y 1995, *J. Knot Theory Ramifications* **4**(2) 189–96.
[5] Doi M and Edwards S F 1986, *The theory of polymer dynamics*, Oxford University Press.
[6] Flory J P 1953, *Principles of Polymer Chemistry*, Cornell University Press.
[7] Gennes P G de 1979, *Scaling Concepts in Polymer Physics*, Cornell University Press.
[8] Holmes V F and Cozzarelli N R 2000, *Proc Natl Acad Sci USA* **97** 1322-4.
[9] Klenin K V, Vologodskii A V, Anshelevich V V, Dykhne A M and Frank-Kamenetskii M D 1988, *J Biomolec Str and Dyn* **5** 1173–85.
[10] Marenduzzo D et al 2009, *Proc Natl Acad Sci USA* **106** 22269–74.
[11] Michletti C, Marenduzzo D, Orlandini E and Sumners D W 2008, *Biophys J* **95** 3591–9.
[12] Millett K 2000, *Knots in Hellas'98 (Delphi), Series on Knots and Everything* **24**, World Scientific 306–34.
[13] Plunkett P et al 2007, *Macromolecules* **40** 3860–7.
[14] Rayleigh L 1919, *Phil. Mag. S. 6.* **37**(220) 321–47.
[15] Varela R, Hinson K, Arsuaga J and Diao Y 2009, J. Phys. A: Math. Theor. **42** 095204.