J Stat Phys (2011) 143: 102–138
DOI 10.1007/s10955-011-0164-4

The Generation of Random Equilateral Polygons

**Sotero Alvarado · Jorge Alberto Calvo ·
Kenneth C. Millett**

Received: 1 November 2010 / Accepted: 21 February 2011 / Published online: 15 March 2011

Abstract Freely jointed random equilateral polygons serve as a common model for polymer rings, reflecting their statistical properties under theta conditions. To generate equilateral polygons, researchers employ many procedures that have been proved, or at least are believed, to be random with respect to the natural measure on the space of polygonal knots. As a result, the random selection of equilateral polygons, as well as the statistical robustness of this selection, is of particular interest. In this research, we study the key features of four popular methods: the Polygonal Folding, the Crankshaft Rotation, the Hedgehog, and the Triangle Methods. In particular, we compare the implementation and efficacy of these procedures, especially in regards to the population distribution of polygons in the space of polygonal knots, the distribution of edge vectors, the local curvature, and the local torsion. In addition, we give a rigorous proof that the Crankshaft Rotation Method is ergodic.

Keywords Knot space · Probability of knotting · Monte Carlo method · Pivot method · Polygonal folding · Crankshaft rotations · Hedgehog method · Triangle method

Electronic supplementary material The online version of this article (doi:10.1007/s10955-011-0164-4) contains supplementary material, which is available to authorized users.

S. Alvarado
Arizona Western College, Yuma, AZ 85365, USA
e-mail: sotero.alvarado@azwestern.edu

J.A. Calvo (✉)
Ave Maria University, Ave Maria, FL 34142, USA
e-mail: jorge.calvo@avemaria.edu

K.C. Millett
University of California, Santa Barbara, CA 93106, USA
e-mail: millett@math.ucsb.edu

1 Introduction

Polymer chains have been modeled as freely jointed random polygons under certain conditions [10, 15, 18, 21, 29, 55, 60, 62, 63, 66]. This simple representation of polymeric chains derives from their statistical properties under theta conditions, in which the effects of excluded volume have been eliminated [9, 10, 26]. The physical behavior of linear polymers under theta conditions is relatively well-understood and extensively explored by numerical simulations and in experiments [10, 71]. The situation for polymer rings, however, becomes more complicated when the topology of the polymer ring is taken into account. For example, one might be interested in how the average behavior of circular polymers of a certain knot type compares with that of another knot type, or with the average properties of the entire population of polymers. In this setting, there is also a growing body of experimental [1–3, 22, 23, 64, 70] and numerical research [14, 15, 18, 21, 29, 50, 55, 60, 62, 63, 67].

In this paper, we will focus on the numerical simulation methods used to study three-space models of such polymer rings. Specifically, we will take a close look at four popular approaches for sampling the space of oriented equilateral knots (based at the origin) in $\mathbf{R}^3$, and at how, under normal circumstances, one might ensure that the chosen sample is sufficiently random, as measured by a wide range of geometric characteristics of the polygon. In addition, we will give a rigorous proof that, like the Polygonal Folding Method [46], the Crankshaft Rotation Method [36, 72] also gives a Markov chain Monte Carlo algorithm where the Markov chain has a unique limit distribution which is uniform. Some of these methods are, for the most part, inspired by methods employed in the study of self-avoiding polygons in the cubic lattice where one wishes to construct an ergodic Markov chain in order to characterize, for example, the frequency and complexity of knotting [58]. The Polygonal Folding Method is a three-space translation of the pivot algorithm employed in the study of self-avoiding walks [37, 40, 41]. The Crankshaft Rotation Method is, similarly, analogous to the pivot algorithm although quite different from the Polygonal Folding Method. Each randomly selects a pair of vertices that separate the polygon into two components, only one of which is then modified in a random structure-preserving manner. The lattice models have been much more amenable to rigorous theoretical and analytical study than the continuum models that are the subject of this paper.

For the *space* $\mathrm{K}(n)$ *of equilateral polygonal knots* with n edges, we take the collection of n-edge polygons in three-dimensional Euclidean space whose edge length is one and for which we have selected an initial vertex and an orientation or direction along the vertices. See Fig. 1. A translation of three-space allows us to assume that the initial vertex is situated at the origin of Euclidean space, so that we might characterize a polygon by its edge vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n$ leading from the initial vertex to the second, from the second vertex to the third, from the third vertex to the fourth, and so on. This sequence of edge vectors corresponds to a point in the product space of n two-dimensional spheres:

$$(\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n) \in \mathbf{S}^2 \times \mathbf{S}^2 \times \cdots \times \mathbf{S}^2.$$

The closure of the space of equilateral polygons is the subset $\mathbb{K}(n)$ of this product on which the sum of the edge vectors is the zero vector:

$$\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \cdots + \mathbf{e}_n = \mathbf{0} = (0, 0, 0).$$

$\mathbb{K}(n)$ is endowed with a natural measure inherited from the product measure on $\mathbf{S}^2 \times \cdots \times \mathbf{S}^2$. The result is the $(2n-3)$-dimensional set of polygons whose vertices are defined by the edge

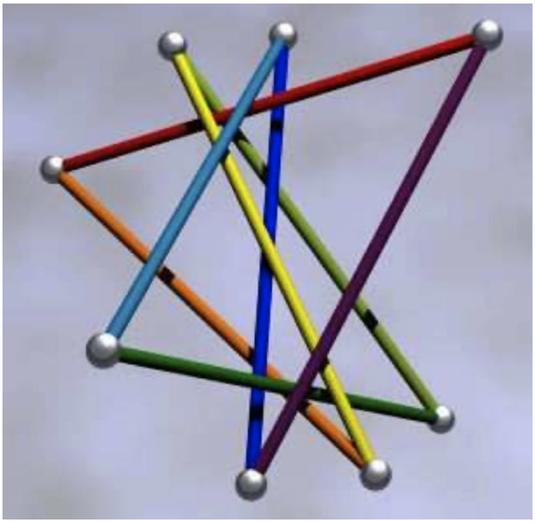


Fig. 1 (Color online) An eight-edge equilateral 8_{19} knot. (Image provided by Peterson Trethewey)

vectors as

$$
\begin{aligned}
v_0 &= \mathbf{0}, \\
v_1 &= \mathbf{e}_1, \\
v_2 &= \mathbf{e}_1 + \mathbf{e}_2, \\
v_3 &= \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3, \\
&\vdots \\
v_{n-1} &= \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \cdots + \mathbf{e}_{n-1}.
\end{aligned}
$$

The space of equilateral polygonal knots $\mathrm{K}(n)$ is the open dense subset of $\mathbb{K}(n)$ defined by the requirement that the polygons be topologically embedded, that is, that two edges either intersect exactly at their common endpoint (assuming they are adjacent) or not at all. From $\mathbb{K}(n)$, it inherits the structure of a $(2n-3)$-dimensional manifold. The connected components of $\mathrm{K}(n)$ are defined to be the *geometric knot types* of our space. In this work, however, we will be concerned with the *topological knot types* defined by the requirement that there be an orientation-preserving homeomorphism of three-space taking one polygonal configuration to the other.

The complement of $\mathrm{K}(n)$ is the set of measure zero in $\mathbb{K}(n)$ consisting of the singular images of polygons. When sampling uniformly with respect to the measure, these singular polygons will occur with probability zero. Thus the knot type of a polygon in $\mathbb{K}(n)$ will be defined with probability one. Furthermore, as the Hedgehog and Triangle methods employ singular configurations as starting points, this shows that these initial configurations are converted, with probability one, into non-singular configurations after sufficiently many polygonal folds or crankshaft rotations. In particular, proving that these operations define ergodic Markov chains is critical to ensuring that the resulting data reflects the actual measure structure of the space of polygons.

When our equilateral polygons have three, four, or five edges, the corresponding space $\mathrm{K}(n)$ consists of a single component containing only the trivial *unknot*, a polygon which is both geometrically and topologically equivalent to a circle. At six edges, the trefoil knot 3_1 makes its first appearance. In fact, Calvo [4] proved that K(6) contains five components: one of unknots and two for the left- and right-handed trefoils, respectively. Calvo also showed that there are at least five components in K(7): one each for the trivial knot, the left-handed trefoil, and right-handed trefoil, and two for the figure-eight knot 4_1. For K(8), where the situation is even more complex, Calvo showed that at most twenty distinct topological knot types (taking chirality into account) were possible. In 2006, Millett confirmed the final case by constructing the equilateral eight-edge 8_{19} knot with vertices

$$\begin{aligned}
v_0 &= (0.0, 0.0, 0.0),\\
v_1 &= (4.24498869823336, -5.2980622416122225, 4.2320925599369605),\\
v_2 &= (1.6213894406256604 - 2.44655836102984, -2.7668825474986716),\\
v_3 &= (6.955383417168724, -3.144912356769294, 3.154334384419697),\\
v_4 &= (-0.5839489586644555, -5.8100837980223154, 2.91911424190918),\\
v_5 &= (3.927828645068674, -2.44655836102984, -2.7668825474986716),\\
v_6 &= (2.059118770743978, -5.191514251536551, 4.511382394420769),\\
v_7 &= (5.379505488302319, -5.235005412573363, -2.7668825474986716),
\end{aligned}$$

shown in Fig. 1. Despite intense Monte Carlo and constructive efforts, the existence of this equilateral knot has otherwise escaped detection, a fact making the limitations of both of these approaches concrete.

One commonly-used strategy for identifying equilateral knots with small numbers of edges employs a random search of knot space. Of course, for nine edges and above, where one does not know exactly which knots can be realized as polygons, this strategy cannot yield a demonstrably complete enumeration of equilateral polygonal knots. Nevertheless, we do know that the number of topologically distinct knots increases exponentially as a function of the number of edges, and that the knots observed become more and more complex by any of the standard measures. Furthermore, Rawdon and Scharein [61] have developed a variety of perturbative methods that allow them to take a polygonal knot with unequal edge lengths and, simultaneously, reduce the number of edges and make the edge lengths very nearly equal. Using a result from [49], they are then able to deduce from this numerical approximation the existence of a large number of truly equilateral knots. This method has been particularly fruitful in the area of nine-edge polygons, in which any new polygonal knot realization must necessarily be minimal, thanks to the complete enumeration of knot types possible with eight or fewer edges.

The purpose of the research presented here is to study the properties of four attractive Markov chain random walk and Markov chain Monte Carlo algorithms used to generate large samples from knot space. These algorithms, which we describe in Sect. 2, can be used to develop estimates of unknown topological knot populations with a fixed number of edges, as well as of their relative probability distribution functions. In Sect. 3, we give a rigorous foundation for the use of one of these algorithms, the Crankshaft Rotation Method. Then, in Sect. 4, we describe six different numerical experiments concerning the implementation and efficacy of these algorithms.

2 Random Walk and Monte Carlo Algorithms

2.1 The Polygonal Folding Method

The *Polygonal Folding Method* **PFM** [46, 47] may take as its initial state any equilateral polygon in $\mathrm{K}(n)$. Often this is taken as the standard regular n-edge planar polygon. An elementary *polygonal fold* begins by randomly selecting a pair of non-adjacent vertices in the polygon. As long as the vertices are distinct, they determine a folding axis that separates the polygon into two pieces, at least one of which does not contain the base vertex at the origin. An angle between 0 and 2π radians is then randomly selected, and the piece of the polygon not containing the base vertex is folded (or rotated) by the selected angle about the given axis. Figure 2 shows the effects of one such polygonal fold. In each frame of this figure, the dotted line shows the axis of folding, while the dashed lines indicate the original position of the polygon. Animation 1 shows an animated version of this figure; please refer to the Electronic Supplemental Material for more details.[1]

The Polygonal Folding Method has been used to study the energy, rope length, and other spatial properties of random equilateral knots [5, 6, 16, 18, 21, 46–49, 51, 60, 62, 63]. Theorem 1, originally proven in 1994 by Millett [46], implies that the PFM is a Markov chain random walk. In 1996, Kapovich and Millson [34] gave an independent proof of this result, and, in 2005, Toussaint [69] corrected a gap in Millett's original proof, extending the result to higher dimensions and showing that the algorithm requires only $O(n)$ time per fold.

Theorem 1 ([34, 46, 69]) *There exists a finite sequence of elementary polygonal folds taking any equilateral polygon in* $\mathrm{K}(n)$ *to the standard regular planar polygon.*

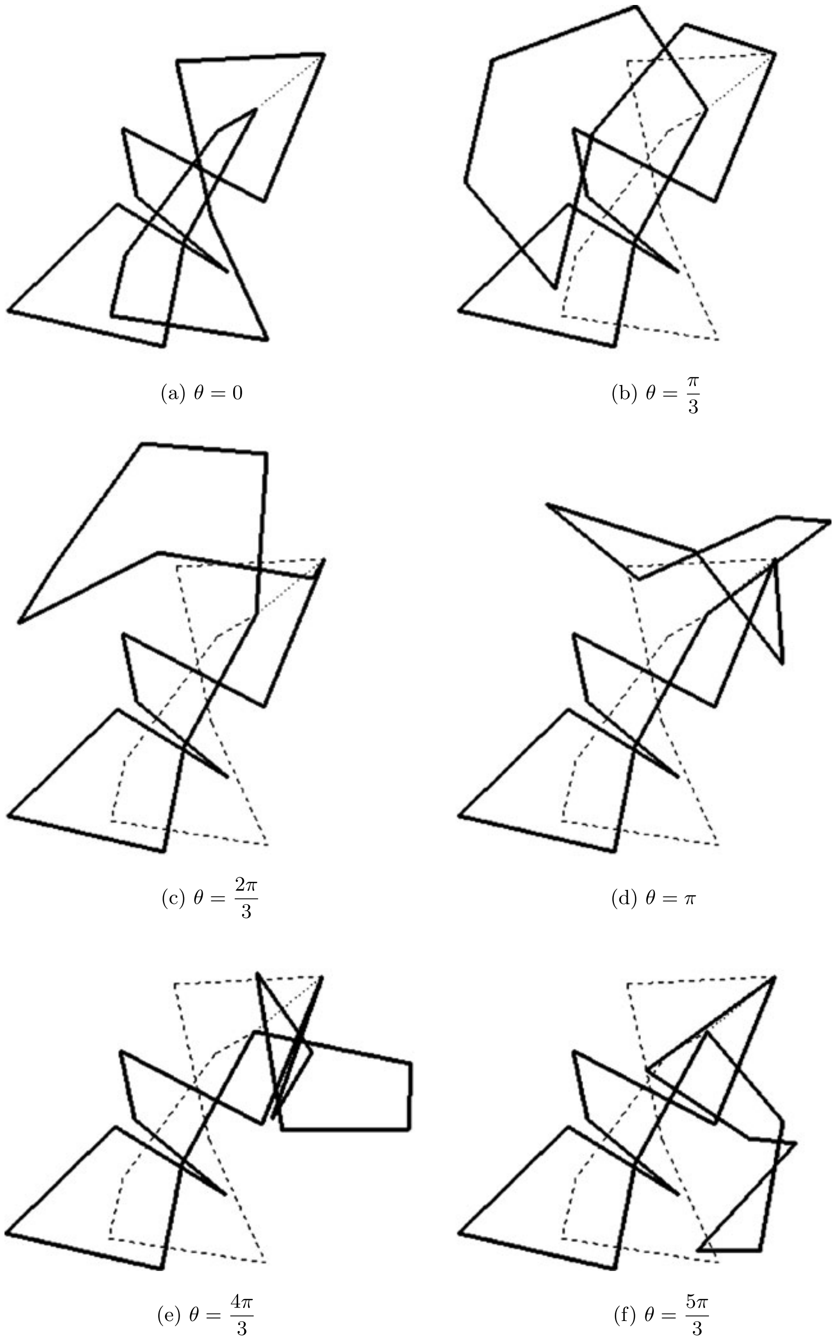
Note that this theorem does not imply that either the geometric or topological knot type of a polygon is preserved, as a single elementary polygonal fold may pass through a number of singular positions thereby repeatedly changing knot types. In fact, although it is natural to apply PFM to embedded polygons in $\mathrm{K}(n)$, it may also be applied to singular polygons in $\mathbb{K}(n)$, as long as one verifies at each step that the axis of folding is well-defined. Typically, this is done by restricting our attention only to those polygons in $\mathbb{K}(n)$ for which all n vertices are distinct.

In order to accelerate the effectiveness of the mixing, and the convergence of measured quantities of configurations, it is common to invoke a Metropolis sampling strategy [43] in which some number of elementary polygonal folds are taken before the next sample is collected [16, 49]. This strategy reduces the correlation between the various samples taken, and ensures that our sampling of knot space creates a more advantageous Markov chain Monte Carlo algorithm. In addition, often one employs a simulated annealing strategy in conjunction with the Metropolis sampling in implementation of the polygonal fold algorithm [7, 16, 35, 49]. In Sect. 4.2, we will study the *sampling rate* (in other words, the number of elementary steps performed between samples) required to minimize, as much as possible, the correlation between the edge vectors of successive samples.

2.2 The Crankshaft Rotation Method

Another strategy, originally proposed by A.M. Dykhne [36, 72] as a constituent of the Hedgehog Method (see Sect. 2.3), is the *Crankshaft Rotation Method* **CRM**. One begins with an equilateral polygon in $\mathrm{K}(n)$, or more generally in $\mathbb{K}(n)$, having edge vectors

[1] Animations 1 and 2 can also be viewed online at www.avemaria.edu/calvopubs.



(a) $\theta = 0$

(b) $\theta = \frac{\pi}{3}$

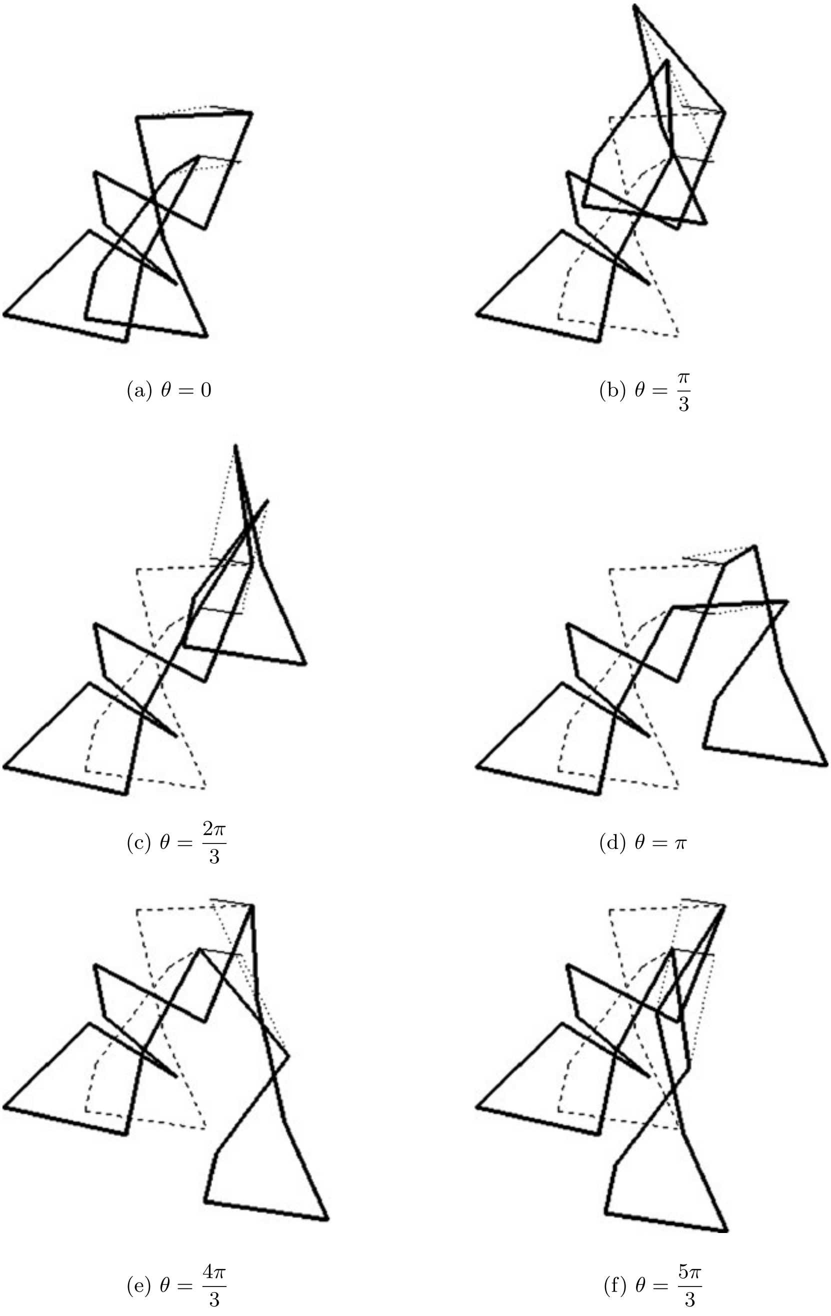
(c) $\theta = \frac{2\pi}{3}$

(d) $\theta = \pi$

(e) $\theta = \frac{4\pi}{3}$

(f) $\theta = \frac{5\pi}{3}$

Fig. 2 Various stages of the Polygonal Folding Method. In each frame, the *dotted line* shows the axis of folding, while the *dashed lines* indicate the original position of the portion of the polygon which rotates about this axis. Note that this motion modifies every edge vector in the rotating portion of the polygon. Refer to Animation 1 in the Electronic Supplemental Material to see additional frames



(a) $\theta = 0$

(b) $\theta = \frac{\pi}{3}$

(c) $\theta = \frac{2\pi}{3}$

(d) $\theta = \pi$

(e) $\theta = \frac{4\pi}{3}$

(f) $\theta = \frac{5\pi}{3}$

Fig. 3 Various stages of the Crankshaft Rotation Method. In each frame, the *dashed lines* indicate the original position of the portion of the polygon which is moved by this rotation. The *thin solid lines* show the components of each rotating edge in the direction of the axis of rotation. The *dotted lines* show their components perpendicular to the axis of rotation. Observe that, with the exception of the two rotating edges, all other edge vectors remains fixed throughout the motion. Refer to Animation 2 in the Electronic Supplemental Material to see additional frames

$(\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n)$. An elementary *crankshaft rotation* consists of the random selection of two non-parallel edge vectors $\mathbf{e}_j$ and $\mathbf{e}_k$. These are then simultaneously rotated by a randomly chosen angle θ between 0 and 2π about the axis determined by $\mathbf{e}_j + \mathbf{e}_k$. In particular, this rotation will take

$$\begin{aligned} \mathbf{e}_j \quad &\mapsto \quad crm(\mathbf{e}_j, \mathbf{e}_k, \theta) = \frac{\mathbf{e}_j + \mathbf{e}_k}{2} + \frac{\mathbf{e}_j - \mathbf{e}_k}{2}\cos\theta + \frac{\mathbf{e}_j \times \mathbf{e}_k}{\|\mathbf{e}_j + \mathbf{e}_k\|}\sin\theta, \\ \mathbf{e}_k \quad &\mapsto \quad crm(\mathbf{e}_k, \mathbf{e}_j, \theta) = \frac{\mathbf{e}_k + \mathbf{e}_j}{2} + \frac{\mathbf{e}_k - \mathbf{e}_j}{2}\cos\theta + \frac{\mathbf{e}_k \times \mathbf{e}_j}{\|\mathbf{e}_k + \mathbf{e}_j\|}\sin\theta. \end{aligned}$$

Figure 3 shows the effects of a crankshaft rotation. In each frame of this figure, the dashed lines indicate the original position of the polygon. In addition, the thin solid lines show the components of each rotating edge in the direction of the axis of rotation, while the dotted lines show their components perpendicular to the axis of rotation. Animation 2 shows an animated version of this figure; please refer to the Electronic Supplemental Material for more details.

Note that this rotation is defined as long as $\mathbf{e}_j$ and $\mathbf{e}_k$ are not parallel. Otherwise, either $\mathbf{e}_j = \mathbf{e}_k$ and the rotation does not alter the polygon, or else $\mathbf{e}_j + \mathbf{e}_k = \mathbf{0}$ and the axis of rotation is not well-defined. Of course, these cases are easily avoided outside a set of measure zero in $\mathbb{K}(n)$. In particular, in our implementations, we need only verify that the two selected edge vectors are not parallel before proceeding with an elementary crankshaft rotation.

In Sect. 3, we prove that the CRM is ergodic, and therefore determines a Markov chain random walk. This result provides the rigorous theoretical foundation for the broad use of the CRM in the generation of equilateral polygons. In particular, when coupled with a Metropolis sampling strategy in which some number of elementary crankshaft rotations are taken before the next sample is collected, the CRM becomes a Markov chain Monte Carlo algorithm where the Markov chain has a unique limit distribution which is uniform. In Sect. 4.2, we will study the sampling rate required to minimize, as much as possible, the correlation between the edge vectors of successive samples.

2.3 The Hedgehog Method

The *Hedgehog Method* [36, 72] is a modification of the PFM and CRM used to generate equilateral polygons with an even number of edges. Instead of starting with the standard regular polygon, one first selects m unit vectors at random. This collection is then doubled in size by including the negatives of these initial vectors, so as to create a set of $n = 2m$ unit vectors whose sum is automatically equal to zero. If we view all of these edge vectors as starting at a common origin then, indeed, we obtain an object which roughly looks like a hedgehog, as shown in Fig. 4(a). Having accomplished this, a number of elementary polygonal folds or crankshaft rotations are then invoked to ensure that the resulting edge vectors are as independent of each other as possible before selecting a representative and starting over with the selection at random of another set of m unit vectors.

The implementation of the Hedgehog Method poses two important problems. First of all, before any folds or rotations are applied, one might require that the polygon corresponding to the hedgehog of n unit vectors have n distinct vertices. This is particularly important when using polygonal folds to do our sampling, since these require us to select at random two distinct vertices to determine the axis of folding.[2] One strategy employed to achieve this setup

[2] Of course, if one can determine that the two randomly-selected vertices are distinct, then one can proceed with the polygonal fold without first requiring that the rest of the vertices of the polygon are distinct. Similarly,

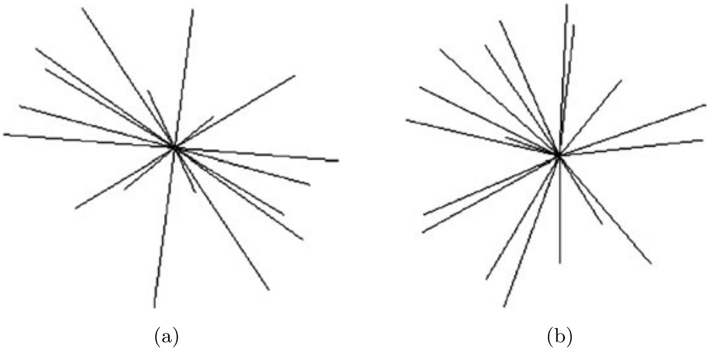


Fig. 4 Two sets of 18 unit vectors produced by (**a**) the Hedgehog Method and (**b**) the Triangle Method. Observe that the symmetry intrinsic to the Hedgehog Method is easier to see than that produced by the Triangle Method

is to permute the collection of n edge vectors until the sum of every proper subsequence of edge vectors is non-zero:

$$\mathbf{e}_i + \mathbf{e}_{i+1} + \cdots + \mathbf{e}_{j-1} + \mathbf{e}_j \neq \mathbf{0} \quad \text{whenever } 1 \leq i \leq j < n.$$

As we observe in Sect. 4.1, this requires, on average, $1.56n$ transpositions.

Secondly, the number of elementary polygonal folds or crankshaft rotations employed to guarantee the independence of the edges has varied widely throughout the literature. For example, [36, 72] use up to $30n$ crankshaft rotations, while [60] propose that roughly $3n$ are sufficient. In each of these cases, the stated objective is to eliminate the correlation between the edge vectors with the ultimate objective of randomly sampling the space of polygons. Observe that, if one insures distinct vertices in the implementation of polygonal folds or crankshaft rotations, the elimination of edge correlations will produce (with probability one) an embedded polygon. Thus, one could dispense with the preparation of an embedded hedgehog polygon and simply be careful to avoid singularities in the implementation of the folds or crankshafts. In Sect. 4.2, we will compare the Pearson Correlation Index [38] between edge vectors (as a function of the sampling rate) to the observed correlation between two independent sets of n random unit vectors.

2.4 The Triangle Method

The *Triangle Method* [53, 55–57] is a modification of the Hedgehog Method in which the initial state (originally consisting of pairs of random unit vectors and their negatives) is replaced by triples of vectors taken from randomly selected equilateral triangles. Thus, an initial ordered set of m vectors is selected. Next, a second ordered set of m vectors, each orthogonal to the corresponding vector in the first set, is selected. Each of these pairs of orthogonal vectors is then used to determine a random equilateral triangle with unit-length edges. The edges of each of each of these triangles are collected together to form a set of $n = 3m$ unit vectors whose sum again equals zero. Observe that while one can visually identify

we can perform a crankshaft rotation by first verifying that the two edge vectors selected are not parallel, regardless of whether the same holds for all the other edge vectors.

the antipodal symmetry of the vectors generated by the Hedgehog Method in Fig. 4(a), the symmetry intrinsic to the Triangle Method is more difficult to see in Fig. 4(b).

As in the case of the Hedgehog Method, this initial state is followed by some number of elementary polygonal folds or crankshaft rotations (assuming that the conditions permit them), thus reducing as much as possible any correlations between the edge vectors. In Sect. 4.2, we will test how the number of iterations affects the value of the Correlation Index, as compared to the value observed between two independent sets of n random unit vectors.

Before applying any polygonal folds or crankshaft rotations, however, we might desire to begin with a polygon having n distinct vertices. In particular, this setup simplifies the implementation of any subsequent polygonal folds. Therefore, we may first choose to permute the sequence of initial edge vectors, paying special attention to sufficiently randomize this sequence so that vectors from the same triangle do not finish in adjacent positions. In particular, we wish to ensure that the turning angles are no longer $\frac{2\pi}{3}$, as they are in an equilateral triangle. As we shall see in Section 4.1, on average, about $0.77n$ random transpositions are required in order to produce a polygon with n distinct vertices. While the resulting configurations continue to have, now hidden, correlations among their edge vectors, Moore and Grosberg [53, 54] have studied the abundance of knotting to confirm the adequacy of this method, at least for their purposes, stating:

> We therefore take special care to compare the results of this method with the unbiased generation using the conditional probability method for both Gaussian distributed and equal length models. We found that no appreciable deviations in knot abundance data arise from the imperfection of the Triangle Method. ([54], p. 9085)

Thus, it seems that the original Triangle Method, without the use of either elementary polygonal folds or crankshaft rotations, may be adequate for some purposes.

3 Ergodicity of the Crankshaft Rotation Method

In the previous section, we described how the Crankshaft Rotation Method has been a central feature of the Hedgehog Method since its inception [36, 72]. Here we will provide the rigorous theoretical foundation to support its widespread application by proving that the action of crankshaft rotations on $\mathrm{K}(n)$ is ergodic. We do this by showing that any point in $\mathrm{K}(n)$ can be deformed by crankshaft rotations into a standard position, namely one corresponding to a planar regular n-gon. To begin, observe that, according to the formulas in Sect. 2.2, a crankshaft rotation of two unit vectors through an angle of π reverses the roles of the two vectors:

$$crm(\mathbf{e}_j, \mathbf{e}_k, \pi) = \mathbf{e}_k \quad \text{and} \quad crm(\mathbf{e}_k, \mathbf{e}_j, \pi) = \mathbf{e}_j.$$

We shall call this type of rotation a *crankshaft flip*.

Lemma 1 *Any point* $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \ldots, \mathbf{e}_n)$ *in* $\mathrm{K}(n)$ *can be deformed by a sequence of crankshaft rotations until it corresponds to a convex planar polygon.*

Proof Let P be a plane in $\mathbb{R}^3$ passing through the origin with unit normal vector $\mathbf{u}$, and suppose that $(\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n)$ does not already lie in P. Since an appropriate sequence of crankshaft flips will permute these vectors, we may assume that $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \ldots, \mathbf{e}_{j-1}$ lie in P and that $\mathbf{e}_j \cdot \mathbf{u}$ is, say, positive. By definition,

$$\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \cdots + \mathbf{e}_n = 0$$

so there is some other vector $\mathbf{e}_k$ in this list such that $\mathbf{e}_k \cdot \mathbf{u}$ is negative. Since a crankshaft flip of $\mathbf{e}_j$ and $\mathbf{e}_k$ will reverse the sign of $\mathbf{e}_j \cdot \mathbf{u}$, there must be some value of θ_0 between 0 and π for which

$$crm(\mathbf{e}_j, \mathbf{e}_k, \theta_0) \cdot \mathbf{u} = 0.$$

After performing this crankshaft rotation, $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \ldots, \mathbf{e}_{j-1}$ and the new $\mathbf{e}_j$ all lie in the plane P. Proceeding inductively, we can eventually place all n vectors in P. Then the vectors can be permuted by a sequence of crankshaft flips so that the (oriented) angles between $\mathbf{e}_1$ and $\mathbf{e}_i$ form an increasing sequence; this will make the polygon convex as desired. □

Note that Lemma 1 can be used to place a polygon, or any part of a polygon, inside any plane P through the origin. In particular, if the first j vectors already lie in P, then they need not be altered in the transformation. Unfortunately, this lemma does not quite hit the mark, since it does not guarantee that the convex planar polygon at which we arrive is regular. For this, we shall need the following result concerning the geometry of regular planar polygons.

Lemma 2 *Suppose that* $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \ldots, \mathbf{e}_n)$ *corresponds to a regular planar polygon. See Fig. 5. Then,*

(a) *the angle between the edge vectors* $\mathbf{e}_i$ *and* $\mathbf{e}_{i+1}$ *measures* $\frac{2\pi}{n}$, *and*
(b) *the angle between the edge vector* $\mathbf{e}_{i+1}$ *and the secant*

$$\mathbf{s}_i = \mathbf{e}_{n-i+1} + \mathbf{e}_{n-i+2} + \cdots + \mathbf{e}_{n-1} + \mathbf{e}_n + \mathbf{e}_1 + \mathbf{e}_2 + \cdots + \mathbf{e}_i$$

measures $(2i+1)\frac{\pi}{n}$.

Proof Recall that the interior angles of a simple planar n-edge polygon sum to $(n-2)\pi$. This means that each interior angle in a regular planar polygon measures $\frac{1}{n}(n-2)\pi = \pi - \frac{2\pi}{n}$. Since the angle between $\mathbf{e}_i$ and $\mathbf{e}_{i+1}$ is an exterior angle of the polygon, its measure is $\frac{2\pi}{n}$.

Now consider the $(2i+1)$-edge polygon formed by $\mathbf{e}_{n-i+1}, \mathbf{e}_{n-i+2}, \ldots, \mathbf{e}_{n-1}, \mathbf{e}_n, \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{i-1}$, and $\mathbf{e}_i$, together with the secant $\mathbf{s}_i$. By the argument above, $2i-1$ of the interior angles for this new polygon measure $\pi - \frac{2\pi}{n}$; thus, the measure of each of the interior angles on either end of the secant $\mathbf{s}_i$ is

$$\frac{1}{2}\left((2i-1)\pi - (2i-1)\left(\pi - \frac{2\pi}{n}\right)\right) = (2i-1)\frac{\pi}{n}.$$

The desired angle between $\mathbf{s}_i$ and $\mathbf{e}_{i+1}$ is the sum of the angle between $\mathbf{s}_i$ and $\mathbf{e}_i$ (which is equal to one of these last two interior angles) and the angle between $\mathbf{e}_i$ and $\mathbf{e}_{i+1}$ (which was computed above). Therefore this angle measures

$$(2i-1)\frac{\pi}{n} + \frac{2\pi}{n} = (2i+1)\frac{\pi}{n}$$

as required. □

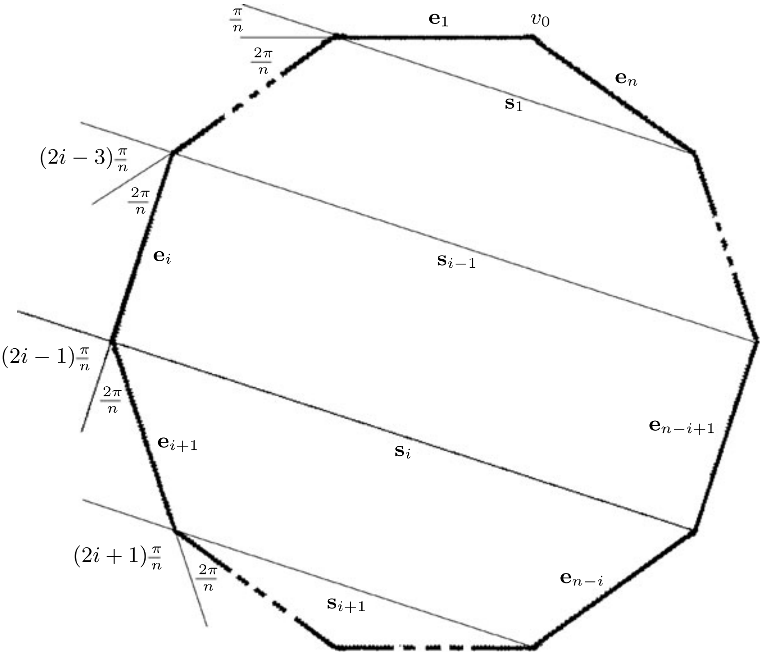


Fig. 5 In a regular planar polygon, each edge vector $\mathbf{e}_i$ makes an angle measuring $\frac{2\pi}{n}$ with the next edge $\mathbf{e}_{i+1}$, and an angle measuring $(2i-1)\frac{\pi}{n}$ with the secant vector $\mathbf{s}_i = \mathbf{e}_{n-i+1} + \mathbf{e}_{n-i+2} + \cdots + \mathbf{e}_{n-1} + \mathbf{e}_n + \mathbf{e}_1 + \mathbf{e}_2 + \cdots + \mathbf{e}_i$

Theorem 2 (Calvo and Millett, 2006) *There exists a finite sequence of pairwise crankshaft rotations taking any equilateral polygon in* K(n) *to the standard regular planar polygon.*

Proof By Lemma 1, we can assume that the collection $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \ldots, \mathbf{e}_n)$ of edge vectors already corresponds to a convex polygon on the xy-plane. We may also assume that this polygon is not already regular. In particular, this means that $n \geq 4$. We shall make our polygon regular in three steps, each of which will require several crankshaft rotations. To simplify our notation, at the outset of each step we shall assume that $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \ldots, \mathbf{e}_n)$ corresponds to a polygon satisfying the properties established in the previous steps.

Step 1. Set the angle between $\mathbf{e}_n$ *and* $\mathbf{e}_1$ *equal to* $\frac{2\pi}{n}$.

Let ϕ_1 be the angle between $\mathbf{e}_n$ and $\mathbf{e}_1$. We can assume (by relabeling if necessary) that $\phi_1 < \frac{2\pi}{n}$. Then

$$\mathbf{e}_1 \cdot \mathbf{e}_n = \cos\phi_1 > \cos\frac{2\pi}{n} \geq \cos\frac{\pi}{2} = 0.$$

Since $\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \cdots + \mathbf{e}_n = 0$, there must be some $\mathbf{e}_k$ for which

$$\mathbf{e}_k \cdot \mathbf{e}_n < 0 = \cos\frac{\pi}{2} \leq \cos\frac{2\pi}{n}.$$

In particular, this means that the angle between $\mathbf{e}_k$ and $\mathbf{e}_n$ is greater than $\frac{2\pi}{n}$. A crankshaft rotation of $\mathbf{e}_1$ and $\mathbf{e}_k$ by an angle of θ will deform the angle $\phi_1(\theta)$ between $\mathbf{e}_n$ and $crm(\mathbf{e}_1, \mathbf{e}_k, \theta)$, taking it from a value less than $\frac{2\pi}{n}$ (when $\theta = 0$) to a value greater than $\frac{2\pi}{n}$ (when $\theta = \pi$). Thus, there is an intermediate value $\theta_0 \in (0, \pi)$ for which $\phi_1(\theta_0) = \frac{2\pi}{n}$.

Let $\tilde{\mathbf{e}}_1 = crm(\mathbf{e}_1, \mathbf{e}_k, \theta_0)$ and $\tilde{\mathbf{e}}_k = crm(\mathbf{e}_k, \mathbf{e}_1, \theta_0)$ denote the first and kth edge vectors after performing this crankshaft rotation, and let P_1 be the plane determined by $\tilde{\mathbf{e}}_\mathbf{1}$ and $\mathbf{e}_n$. We can then apply Lemma 1 and deform the polygon $(\tilde{\mathbf{e}}_1, \mathbf{e}_2, \mathbf{e}_3, \dots, \mathbf{e}_{k-1}, \tilde{\mathbf{e}}_k, \mathbf{e}_{k+1}, \dots, \mathbf{e}_n)$ until it corresponds to a planar polygon in P_1; in particular, we can do this without altering the angle between $\tilde{\mathbf{e}}_\mathbf{1}$ and $\mathbf{e}_n$, which will still measure $\frac{2\pi}{n}$.

Step 2. For each $1 < i \leq \frac{n}{2}$, *make the secant vector*

$$\mathbf{s}_i = \mathbf{e}_{n-i+1} + \cdots + \mathbf{e}_{n-1} + \mathbf{e}_n + \mathbf{e}_1 + \mathbf{e}_2 + \cdots + \mathbf{e}_i$$

parallel to $\mathbf{s}_1 = \mathbf{e}_n + \mathbf{e}_1$.

We begin by assuming that the secants $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{j-1}, \mathbf{s}_j$ are all parallel. Consider the angles ϕ_{j+1} and ϕ_{n-j} between the secant $\mathbf{s}_j$ and each of the vectors $\mathbf{e}_{j+1}$ and $\mathbf{e}_{n-j}$, respectively. If these angles are equal, then $\mathbf{s}_{j+1}$ is already parallel to $\mathbf{s}_j$ and hence to $\mathbf{s}_1$. If not, a crankshaft flip of $\mathbf{e}_{j+1}$ and $\mathbf{e}_{n-j}$ will interchange the relative size of these angles; thus, there is a rotation angle $\theta_0 \in (0, \pi)$ for which the angles between $\mathbf{s}_j$ and each of the vectors $\tilde{\mathbf{e}}_{j+1} = crm(\mathbf{e}_{j+1}, \mathbf{e}_{n-j}, \theta_0)$ and $\tilde{\mathbf{e}}_{n-j} = crm(\mathbf{e}_{n-j}, \mathbf{e}_{j+1}, \theta_0)$ are equal. After this rotation, the new secant vector

$$\tilde{\mathbf{s}}_{j+1} = \tilde{\mathbf{e}}_{n-j} + \mathbf{e}_{n-j+1} + \cdots + \mathbf{e}_{n-1} + \mathbf{e}_n + \mathbf{e}_1 + \mathbf{e}_2 + \cdots + \mathbf{e}_j + \tilde{\mathbf{e}}_{j+1}$$

will be parallel to $\mathbf{s}_1$.

Let P_{j+1} be the plane determined by $\mathbf{s}_j$ and $\tilde{\mathbf{e}}_{j+1}$. Since the secant vectors $\tilde{\mathbf{s}}_{j+1}$ and $\mathbf{s}_j$ are parallel, this plane contains

$$\tilde{\mathbf{e}}_{n-j} = \tilde{\mathbf{s}}_{j+1} - \mathbf{s}_j - \tilde{\mathbf{e}}_{j+1}.$$

Furthermore, the pair of vectors $\mathbf{e}_j$ and $\mathbf{e}_{n-j+1}$ rotate about the axis

$$\mathbf{e}_j + \mathbf{e}_{n-j+1} = \mathbf{s}_j - \mathbf{s}_{j-1}$$

which is parallel to $\mathbf{s}_j$ and therefore lies in P_{j+1}. This pair of vectors can therefore be brought into P_{j+1} by a crankshaft rotation. For similar reasons, the axis of rotation

$$\mathbf{e}_{j-1} + \mathbf{e}_{n-j+2} = \mathbf{s}_{j-1} - \mathbf{s}_{j-2}$$

lies in P_{j+1}, so the pair of vectors $\mathbf{e}_{j-1}$ and $\mathbf{e}_{n-j+2}$ can also be brought into P_{j+1} by a crankshaft rotation. Continuing in this fashion, we can bring each one of the edge vectors $\mathbf{e}_{n-j+1}, \dots, \mathbf{e}_{n-1}, \mathbf{e}_n, \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{j-1}$, and $\mathbf{e}_j$ into P_{j+1}. Then using Lemma 1 we can bring the rest of the polygon into P_{j+1}, all the while keeping the secants $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_j$, and $\tilde{\mathbf{s}}_{j+1}$ parallel to each other.

Proceeding inductively, we can get every single secant $\mathbf{s}_i$ parallel to $\mathbf{s}_1$, as desired.

Step 3. For each $1 \leq i \leq \frac{n}{2}$, *adjust the vectors* $\mathbf{e}_{i+1}$ *and* $\mathbf{e}_{n-i}$ *so the angles they each make with the secant vector*

$$\mathbf{s}_i = \mathbf{e}_{n-i+1} + \cdots + \mathbf{e}_{n-1} + \mathbf{e}_n + \mathbf{e}_1 + \mathbf{e}_2 + \cdots + \mathbf{e}_i$$

measure $(2i+1)\frac{\pi}{n}$.

As above, let ϕ_{i+1} be the angle between $\mathbf{s}_i$ and $\mathbf{e}_{i+1}$; similarly, let ϕ_{n-i} be the angle between $\mathbf{s}_i$ and $\mathbf{e}_{n-i}$. We shall proceed by induction, assuming that

$$\phi_{i+1} = \phi_{n-i} = (2i+1)\frac{\pi}{n}$$

whenever $1 \le i < j$. In particular, this means that the polygon formed by the vectors $\mathbf{e}_{n-j+1}, \mathbf{e}_{n-j+2}, \ldots, \mathbf{e}_{n-1}, \mathbf{e}_n, \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{j-1}, \mathbf{e}_j$, and $\mathbf{s}_j$ agrees in $2j$ consecutive edges with a regular planar n-gon. Furthermore, we shall assume that the polygon determined by the vectors $\mathbf{e}_{j+1}, \mathbf{e}_{j+2}, \ldots, \mathbf{e}_{n-j-1}, \mathbf{e}_{n-j}$, and $\mathbf{s}_j$ is also convex, so that

$$\phi_{j+1} \le \phi_{j+2} \le \cdots \le \phi_{\lfloor \frac{n}{2} \rfloor} \le \pi.$$

Case A. Suppose that $\phi_{j+1} < (2j+1)\frac{\pi}{n}$.

There must be some k with $j+2 \le k \le \frac{n}{2}$ for which $\phi_k > (2k-1)\frac{\pi}{n}$; for otherwise, let $\mathbf{u} = \mathbf{s}_j / \|\mathbf{s}_j\|$ be a unit vector pointing in the same direction as $\mathbf{s}_j$. Then

$$\begin{aligned}
\|\mathbf{s}_j\| &= \mathbf{s}_j \cdot \mathbf{u} \\
&= (-\mathbf{e}_{j+1} - \mathbf{e}_{j+2} - \cdots - \mathbf{e}_{n-j-1} - \mathbf{e}_{n-j}) \cdot \mathbf{u} \\
&= -\mathbf{e}_{j+1} \cdot \mathbf{u} - \mathbf{e}_{j+2} \cdot \mathbf{u} - \cdots - \mathbf{e}_{n-j-1} \cdot \mathbf{u} - \mathbf{e}_{n-j} \cdot \mathbf{u} \\
&= -\cos\phi_{j+1} - \cos\phi_{j+2} - \cdots - \cos\phi_{n-j-1} - \cos\phi_{n-j} \\
&= -\cos\phi_{j+1} - \cos\phi_{j+2} - \cdots - \cos\phi_{\lfloor \frac{n}{2} \rfloor} \\
&\quad\ - \epsilon - \cos\phi_{\lfloor \frac{n}{2} \rfloor} - \cdots - \cos\phi_{j+2} - \cos\phi_{j+1} \\
&< -\cos\left((2j+1)\frac{\pi}{n}\right) - \cos\left((2j+3)\frac{\pi}{n}\right) - \cdots - \cos\left(\left(2\left\lfloor \frac{n}{2} \right\rfloor + 1\right)\frac{\pi}{n}\right) \\
&\quad\ - \epsilon - \cos\left(\left(2\left\lfloor \frac{n}{2} \right\rfloor + 1\right)\frac{\pi}{n}\right) - \cdots - \cos\left((2j+3)\frac{\pi}{n}\right) - \cos\left((2j+1)\frac{\pi}{n}\right),
\end{aligned}$$

where ϵ equals 0 if n is even and 1 if n is odd. The last expression gives the length of the secant joining two vertices separated by $n-2j$ edges in a regular n-gon. Of course, looking at the other side of the regular n-gon, we find that this length is also equal to that of a secant joining two vertices separated by $2j$ edges. Thus, it is equal to $\|\mathbf{s}_j\|$, which is a contradiction. Therefore we can assume that $\phi_k > (2k-1)\frac{\pi}{n}$ for some k.

Now, a crankshaft rotation of $\mathbf{e}_{j+1}$ and $\mathbf{e}_k$ by an angle of θ will deform the angle $\phi_{j+1}(\theta)$ between $\mathbf{s}_j$ and $crm(\mathbf{e}_{j+1}, \mathbf{e}_k, \theta)$, taking it from a value less than $(2j+1)\frac{\pi}{n}$ to a value greater than $(2k-1)\frac{\pi}{n} > (2j+1)\frac{\pi}{n}$. Thus, there is an intermediate value $\theta_0 \in (0, \pi)$ for which $\phi_{j+1}(\theta_0) = (2j+1)\frac{\pi}{n}$.

Let $\tilde{\mathbf{e}}_{j+1} = crm(\mathbf{e}_{j+1}, \mathbf{e}_k, \theta_0)$ and $\tilde{\mathbf{e}}_k = crm(\mathbf{e}_k, \mathbf{e}_{j+1}, \theta_0)$ denote the $j+1$st and kth edge vectors after performing this rotation. By symmetry, a pairwise crankshaft rotation of $\mathbf{e}_{n-j}$ and $\mathbf{e}_{n-k+1}$ by the same angle θ_0 will set ϕ_{n-j} equal to $(2j+1)\frac{\pi}{n}$, so we let $\tilde{\mathbf{e}}_{n-k+1} = crm(\mathbf{e}_{n-k+1}, \mathbf{e}_{n-j}, \theta_0)$ and $\tilde{\mathbf{e}}_{n-j} = crm(\mathbf{e}_{n-j}, \mathbf{e}_{n-k+1}, \theta)$ be the results of this second rotation. Since our rotation angles agree, each of the new secant vectors

$$\tilde{\mathbf{s}}_i = \mathbf{e}_{n-i+1} + \cdots + \tilde{\mathbf{e}}_{n-k+1} + \cdots + \tilde{\mathbf{e}}_{n-j} + \cdots + \mathbf{e}_n + \mathbf{e}_1 + \cdots + \tilde{\mathbf{e}}_{j+1} + \cdots + \tilde{\mathbf{e}}_k + \cdots + \mathbf{e}_i$$

is still parallel to $\mathbf{s}_1$. Let Q_{j+1} be the plane determined by $\mathbf{s}_j$ and $\tilde{\mathbf{e}}_{j+1}$; note that this plane also contains $\tilde{\mathbf{e}}_{n-j} = \tilde{\mathbf{s}}_{j+1} - \mathbf{s}_j - \tilde{\mathbf{e}}_{j+1}$. Furthermore, since the secants $\tilde{\mathbf{s}}_i$ are parallel, we can bring the rest of the polygon (two edges at a time) into Q_{j+1} just as in Step 2. The resulting polygon will then have

$$\phi_{i+1} = \phi_{n-i} = (2i+1)\frac{\pi}{n}$$

whenever $1 \le i < j+1$, and thus agree with a regular n-gon in at least $2j+2$ edges.

Case B. Suppose that $\phi_{j+1} > (2j+1)\frac{\pi}{n}$.

Then, there must be some k with $j+2 \leq k \leq \frac{n}{2}$ for which $\phi_k < (2k-1)\frac{\pi}{n}$; for otherwise, let $\mathbf{u} = \mathbf{s}_j / \|\mathbf{s}_j\|$ be a unit vector pointing in the same direction as $\mathbf{s}_j$. Then

$$\begin{aligned}
\|\mathbf{s}_j\| &= \mathbf{s}_j \cdot \mathbf{u} \\
&= (-\mathbf{e}_{j+1} - \mathbf{e}_{j+2} - \cdots - \mathbf{e}_{n-j-1} - \mathbf{e}_{n-j}) \cdot \mathbf{u} \\
&= -\mathbf{e}_{j+1} \cdot \mathbf{u} - \mathbf{e}_{j+2} \cdot \mathbf{u} - \cdots - \mathbf{e}_{n-j-1} \cdot \mathbf{u} - \mathbf{e}_{n-j} \cdot \mathbf{u} \\
&= -\cos\phi_{j+1} - \cos\phi_{j+2} - \cdots - \cos\phi_{n-j-1} - \cos\phi_{n-j} \\
&= -\cos\phi_{j+1} - \cos\phi_{j+2} - \cdots - \cos\phi_{\lfloor \frac{n}{2} \rfloor} \\
&\quad - \epsilon - \cos\phi_{\lfloor \frac{n}{2} \rfloor} - \cdots - \cos\phi_{j+2} - \cos\phi_{j+1} \\
&> -\cos\left((2j+1)\frac{\pi}{n}\right) - \cos\left((2j+3)\frac{\pi}{n}\right) - \cdots - \cos\left(\left(2\left\lfloor \frac{n}{2} \right\rfloor + 1\right)\frac{\pi}{n}\right) \\
&\quad - \epsilon - \cos\left(\left(2\left\lfloor \frac{n}{2} \right\rfloor + 1\right)\frac{\pi}{n}\right) - \cdots - \cos\left((2j+3)\frac{\pi}{n}\right) - \cos\left((2j+1)\frac{\pi}{n}\right),
\end{aligned}$$

where ϵ equals 0 if n is even and 1 if n is odd. As before, the last expression gives the length of the secant joining two vertices separated by $2j$ edges in a regular n-gon and must therefore equal $\|\mathbf{s}_j\|$. This would be a contradiction.

Applying the same method as in Case A, we can set ϕ_k and ϕ_{n-k+1} equal to $(2k-1)\frac{\pi}{n}$, all the while reducing the size of ϕ_{j+1} and ϕ_{n-j}. If, after this reduction, ϕ_{j+1} is still greater than $(2j+1)\frac{\pi}{n}$, we can repeat Case B. Each time we do so, there will be one fewer value of k for which $\phi_k < (2k-1)\frac{\pi}{n}$, and so this process must eventually terminate with $\phi_{j+1} \leq (2j+1)\frac{\pi}{n}$. We can then perform the rotations prescribed in Case A and obtain a polygon for which

$$\phi_{i+1} = \phi_{n-i} = (2i+1)\frac{\pi}{n}$$

whenever $1 \leq i < j+1$. Again, such a polygon agrees with a regular n-gon in at least $2j+2$ edges.

According to Lemma 2, when Step 3 is completed, we will have a collection of edge vectors $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \ldots, \mathbf{e}_n)$ corresponding to a regular planar polygon, as desired. □

Theorem 2 shows that the random crankshaft rotations are sufficient to define a Markov chain random walk. By adding permutations of the edge vectors to each elementary crankshaft rotation, the rate of mixing inherent in the algorithm is further increased. We can also use a Metropolis sampling strategy by performing some given number of elementary steps before selecting a sample. For instance, in Sect. 4.2, we determine experimentally that performing as few as n random crankshaft rotations between samples will produce sample polygons whose edge vectors are, on average, as uncorrelated as possible according to the Pearson Correlation Index.

In the next section we describe six different experiments forming part of our statistical comparison of the algorithms discussed above. In particular, our investigation looks at issues concerning the implementation of these algorithms, and their application in determining geometric measures such as the total curvature, total torsion, and spherical distribution of edge vectors of a random equilateral polygon, as well as the probability distribution function of knots in the space of equilateral polygons.

4 Experiment Design, Analysis and Conclusions

In this section, we analyze the data gathered in our numerical study of the four methods described in Sect. 2. Our main goal is to illuminate several critical features of both the implementation and the results of these algorithms, and to provide other researchers with information that may assist them in selecting the most appropriate method for their specific application. In our analysis, we propose to explore to the following questions:

1. How must the polygons generated by the Hedgehog or Triangle Methods be modified in order to prevent those singular cases that can interfere with sampling by random polygonal folds or crankshaft rotations?
2. How many iterations of random polygonal folds or crankshaft rotations should be employed in order to minimize the correlation between successive polygon selections, and thereby improve the random sampling of the space of knots?
3. How much time, relatively speaking, does it take to determine successive polygons when one takes into account the number of operations required to initialize a random walk and to accomplish the desired selection?
4. How does the distribution of edge vectors selected by the algorithms compare to that of a random selection of edge vectors without the closure requirement?
5. How well do these algorithms represent the geometry of random polygons as measured by the total curvature and total torsion of the polygons?
6. How well do these algorithms sample the space of equilateral polygonal knots with respect to its partition by topological knot type?

The four polygon generation algorithms were implemented independently in *Mathematica 6.0* and in *Fortran 77*. In both settings, the programs were essentially identical, with only the polygon generation modules being changed. Random numbers were generated using *Mathematica*'s *RandomReal* extended cellular automata-based method and *Fortran*'s random number generator. Seed numbers for each of the programs were provided by Mads Haahr's `random.org` website [31]. For the statistical analysis, we used *Mathematica*'s built-in *Pearson Correlation Index* function and wrote *Mathematica* programs to perform the other statistical tests.

4.1 Preparation of Initial Edge Vectors

As we noted earlier, polygonal folds and crankshaft rotations form the key constituents in the sampling phase of the Hedgehog and the Triangle Methods. Although it is not a strict requirement, most implementations of PFM and CRM require that the starting configuration be a non-singular, generic polygon. In particular, for the PFM, the two randomly-selected vertices must be distinct in order to determine a well-defined axis of folding. Similarly, for the CRM, the two edge vectors selected must be non-parallel. Of course, a polygon need only have one pair of distinct vertices or one pair of non-parallel edges to serve as a suitable initial state for PFM or CRM, respectively, assuming that one checks for this condition every time a random pair of vertices or edges is selected. In general, however, a starting polygon all of whose vertices are distinct and none of whose edges are parallel is preferred.

For the simple versions of the PFM and CRM, in which a regular planar polygon is taken as the initial position, the only problems occur only for an even number of edges. In this case, every pair of opposite edges is parallel, so our implementation of CRM must confirm that these opposite edges are never selected simultaneously.

On the other hand, unless special care is taken, the initial generation phase in both the Hedgehog and Triangle Methods will typically produce a starting polygon which is neither

Table 1 The mean number of transpositions (and their standard deviation) required to achieve non-singular polygons from the Hedgehog and Triangle Methods as a function of the number n of edges

Number of edges	Hedgehog Method		Triangle Method	
$n = 12k$	transpositions	std dev	transpositions	std dev
12	13.61	8.09	5.58	2.79
24	28.54	14.59	12.30	5.08
36	45.05	22.83	20.27	7.50
48	63.02	28.87	28.87	10.22
60	80.52	35.51	37.79	12.71
72	99.81	42.68	47.41	15.29
84	120.42	49.54	57.93	18.49
96	139.77	56.72	67.63	19.69
108	160.53	64.03	77.64	23.03
120	181.37	73.47	87.50	25.14

non-singular nor generic. As a consequence, before applying any PFM's or CRM's, it is typical to randomly permute the initial edge vectors some number of times until a non-singular configuration is achieved.[3] In as much as these transpositions represent an intrinsic overhead in the implementation of the Hedgehog or Triangle Methods, we provide a numerical estimate of the number of transpositions required to transform a polygon generated by each of these methods into one having all distinct vertices.

4.1.1 Computations

A sample set of 1 000 random $12k$-edge polygons was generated for each integer $k = 1, \dots, 10$ using each of the Hedgehog and Triangle Methods. Each polygon in these sample sets then underwent a sequence of edge permutations until a polygon with distinct vertices was produced. The mean number of transpositions required for each sample set is reported in Table 1 and in Fig. 6.

4.1.2 Analysis

Using the least-squares linear regression for the data in Table 1, we determine that, for the Hedgehog Method, roughly $1.56n$ transpositions are required to produce a non-singular n-edge polygon. Similarly, for the Triangle Method, roughly $0.77n$ transpositions are required. Since a closer inspection of the data suggests that the growth may actually be a power function, we also computed the best-fitting power function describing each data set. With an R^2 value of 0.999, the mean number of transpositions necessary for the Hedgehog Method is $0.797n^{1.1309}$ while that for the Triangle function is $0.2715n^{1.207}$. This provides another expression of the increasing cost of data preparation for the Hedgehog Method as compared to the Triangle Method when embedded polygons are required for the initial state.

4.1.3 Conclusions

Our results show that roughly twice as many transpositions are required for the Hedgehog Method than for the Triangle Method. Nevertheless, because of the large standard deviation

[3] Unfortunately, permutations alone will not create the generic polygon required by CRM. For this reason, before each rotation, our implementation of CRM always verifies that the selected edges are not parallel.

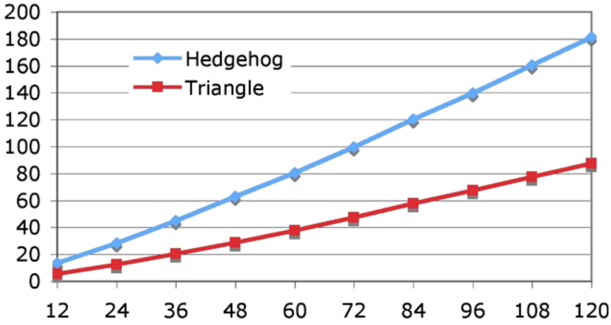


Fig. 6 (Color online) A comparison of the mean number of transpositions required to achieve non-singular polygons from the Hedgehog and Triangle Methods as a function of the number n of edges

for both methods (roughly $0.59n$ and $0.20n$, respectively), one should probably not rely solely on performing $1.56n$ or $0.77n$ transpositions as a means of preventing problems in the implementation of the polygonal folds. Instead, it may be wise to continue the random permutations until one can verify that a configuration with distinct vertices has been reached. Alternatively, it might be prudent to check that conditions permitting the implementation of a polygonal fold are satisfied at each step.

4.2 Determination of the Sampling Rate

The principal objective of all of the algorithms described in Sect. 2 consists of generating a collection of polygons which randomly sample the space of knots and, therefore, are as different as possible from each other. In particular, the edge vectors of each polygon should be as uncorrelated as possible with those of the preceding polygon. This is accomplished by means of a Metropolis sampling method in which the polygonal folding or the crankshaft rotation steps are repeated a specified number of times prior to the selection of the next sample polygon. This number of iterations, also known as the sampling rate, is typically given as a function of the number of edges in the class of polygons being studied. This sampling rate is a principal contributor to the complexity of the method, i.e. the number of operations that are required to generate a random sampling of the space of knots.

In the literature, there appears to be a considerable range of views on how large this sampling rate ought to be. Thus, we now compare the effect of using various sampling rates on the Pearson Correlation Index (PCI) between each successive pair of polygons sampled.

4.2.1 Computations

The Pearson Correlation Index (also known as the Pearson Product-Moment Correlation) provides a measure of the degree to which two sets of data, **X** and **Y**, are linearly related [38]. It is defined as the covariance between the two sets of data divided by their standard deviations; in other words,

$$PCI(\mathbf{X},\mathbf{Y}) = \frac{\sum_{i=1}^{n}(x_i - \overline{\mathbf{X}})(y_i - \overline{\mathbf{Y}})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{\mathbf{X}})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{\mathbf{Y}})^2}},$$

where $\overline{\mathbf{X}}$ and $\overline{\mathbf{Y}}$ denote the mean of **X** and **Y**, respectively. Thus, $PCI(\mathbf{X},\mathbf{Y})$ takes on a value in the closed interval $[-1, 1]$; values near ± 1 indicate a strong linear relationship between the data sets, while values near 0 indicate a weak relationship. In our case, we shall determine the correlation between two n-edge polygons by comparing the sequences of $3n$

coordinates obtained from the n edge vectors. We calculate the PCI for these sequences n times, once for each choice of starting vertex, taking as our final result the maximum absolute value observed.

First, two sample sets of 1 000 random $12k$-edge polygons were generated for each integer $k = 1, \ldots, 10$ by repeatedly applying the PFM and the CRM, respectively, to a standard $12k$-edge regular polygon. After every $n = 12k$ iterations, the Pearson Correlation Index between the new and the initial polygons was computed. In addition, after every $5n = 60k$ iterations, the resulting polygon was taken as the initial polygon for the next round of PFM's or CRM's. The whole process was repeated 1 000 times. The mean PCI values for these sample sets are recorded in Tables 2 and 3, respectively. These values give a measure of the relative efficacy between the PFM and the CRM under the various sampling rates tested (n, $2n$, $3n$, $4n$, and $5n$).

Next, two new sample sets of 1 000 random $12k$-edge polygons were generated for each integer $k = 1, \ldots, 10$ using the Hedgehog Method. In each case, a sequence of edge permutations was used to ensure the non-singularity of the resulting polygons as described in Sect. 4.1. Each polygon in the first of these sets then underwent a sequence of polygonal folds, while each polygon in the second set underwent a sequence of crankshaft rotations. In each case, the Pearson Correlation Index between the new and the initial polygons was computed every $n = 12k$ iterations. The mean PCI values for these polygons appear in Tables 4 and 5.

Finally, two more sample sets of 1 000 random $12k$-edge polygons were produced for each integer $k = 1, \ldots, 10$ using the Triangle Method. As in the previous case, a sequence of edge permutations was used to ensure the non-singularity of the resulting polygons. Then polygons in the first of these sets were subjected to polygonal folds, while polygons in the second set were subjected to crankshaft rotations. The Pearson Correlation Index between the new and the initial polygons was computed every $n = 12k$ iterations. The mean PCI values for these polygons are reported in Tables 6 and 7. In order to establish a benchmark for the data above, we calculated the Peterson Correlation Index between two random sequences of $12k$ unit vectors (without any closure condition) for each integer $k = 1, \ldots, 10$. This benchmark value was computed 1 000 times and its mean appears in the column labeled "random" in Tables 2 through 7.

4.2.2 Analysis

For each of the six scenarios outlined above, we consider the correlation between the edge vectors (as measured by the average PCI values) as a function of the number of iterations of polygonal folds or crankshaft rotations performed (in other words, as a function of the sampling rate). These values are then compared to the mean PCI calculated for the corresponding set of random edge vectors without the closure requirement, as reported in the "random" column of Tables 2 through 7. For each value of n, the PCI that comes closest to this "random" mean correlation is indicated in bold font.

In many cases, the computed PCI values were actually lower than the mean correlation associated with a set of random edge vectors. This might lead one to erroneously conclude that the resulting polygon is somehow "more random" than a random collection of vectors. Indeed, with the exception of the data presented in Table 2, all of the computed PCI values lie within one standard deviation of the mean "random" correlations. Whereas this indicates that the differences among this data should not be considered statistically significant, it also points to the fact that all six scenarios are quite effective at reducing edge correlations, even when the sampling rate used is relatively small. At any rate, we shall assume that any PCI

Table 2 Edge vector correlations (PCI) for the Polygonal Folding Method using various sampling rates as a function of the number n of edges. The mean PCI values (and their standard deviation) between two random sequences of n unit vectors without any closure condition are given in the columns labeled "random" to provide a benchmark for comparison. For each value of n, the PCI that comes closest to this random mean correlation is indicated in bold font

Edges	Polygonal folds					Random	
$n = 12k$	n	$2n$	$3n$	$4n$	$5n$	mean	std dev
12	0.3471	**0.3443**	0.2371	0.2274	0.2312	0.3280	0.0828
24	**0.2717**	0.1921	0.1633	0.1571	0.1550	0.2695	0.0605
36	**0.2199**	0.1525	0.1343	0.1306	0.1228	0.2352	0.0484
48	**0.1969**	0.1362	0.1123	0.1081	0.1113	0.2127	0.0413
60	**0.1707**	0.1140	0.0998	0.0949	0.0937	0.1972	0.0360
72	**0.1533**	0.1025	0.0923	0.0881	0.0873	0.1836	0.0319
84	**0.1389**	0.0945	0.0830	0.0845	0.0776	0.1732	0.0289
96	**0.1233**	0.0861	0.0792	0.0768	0.0756	0.1650	0.0269
108	**0.1130**	0.0798	0.0721	0.0706	0.0714	0.1574	0.0247
120	**0.1066**	0.0752	0.0699	0.0672	0.0642	0.1510	0.0236

Table 3 Edge vector correlations (PCI) for the Crankshaft Rotation Method using various sampling rates as a function of the number n of edges. The mean PCI values (and their standard deviation) between two random sequences of n unit vectors without any closure condition are given in the columns labeled "random" to provide a benchmark for comparison. For each value of n, the PCI that comes closest to this random mean correlation is indicated in bold font

Edges	Crankshaft rotations					Random	
$n = 12k$	n	$2n$	$3n$	$4n$	$5n$	mean	std dev
12	0.3457	0.3371	0.3354	0.3377	**0.3330**	0.3280	0.0828
24	0.2737	**0.2665**	0.2648	0.2673	0.2679	0.2695	0.0605
36	**0.2326**	0.2346	0.2339	0.2307	0.2302	0.2352	0.0484
48	**0.2106**	0.2086	0.2106	0.2089	0.2093	0.2127	0.0413
60	**0.1946**	0.1931	0.1919	0.1932	0.1928	0.1972	0.0360
72	**0.1811**	0.1808	0.1786	0.1797	0.1808	0.1836	0.0319
84	**0.1696**	0.1692	0.1692	0.1699	0.1694	0.1732	0.0289
96	**0.1623**	0.1614	0.1628	0.1609	0.1612	0.1650	0.0269
108	**0.1546**	0.1542	0.1554	0.1535	0.1550	0.1574	0.0247
120	**0.1498**	0.1459	0.1480	0.1490	0.1497	0.1510	0.0236

value below the benchmark reflects a level of correlation comparable to that of a random collection of vectors.

4.2.3 Conclusions

The data in Tables 2, 3, and 5 shows that the Polygon Folding Method, the Crankshaft Rotation Method, and the Hedgehog Method with repeated crankshaft rotations can all be expected to achieve a distribution of edge vectors whose correlations are comparable to those of a random collection of vectors by sampling every n iterations (or slightly more when n is small). Similarly, the data in Tables 4 and 6 indicates that a sampling rate of $2n$ iterations is

Table 4 Edge vector correlations (PCI) for the Hedgehog Method with repeated polygonal folds at various sampling rates as a function of the number n of edges. The mean PCI values (and their standard deviation) between two random sequences of n unit vectors without any closure condition are given in the columns labeled "random" to provide a benchmark for comparison. For each value of n, the PCI that comes closest to this random mean correlation is indicated in bold font

Edges	Polygonal folds					Random	
$n = 12k$	n	$2n$	$3n$	$4n$	$5n$	mean	std dev
12	0.3699	0.3440	0.3312	0.3362	**0.3260**	0.3280	0.0828
24	0.2930	**0.2667**	0.2671	0.2641	0.2640	0.2695	0.0605
36	0.2508	**0.2359**	0.2336	0.2265	0.2278	0.2352	0.0484
48	0.2265	**0.2153**	0.2086	0.2057	0.2073	0.2127	0.0413
60	0.2044	**0.1957**	0.1916	0.1883	0.1917	0.1972	0.0360
72	0.1944	**0.1818**	0.1796	0.1771	0.1778	0.1836	0.0319
84	0.1833	**0.1723**	0.1689	0.1679	0.1673	0.1732	0.0289
96	0.1765	**0.1627**	0.1599	0.1605	0.1587	0.1650	0.0269
108	0.1660	**0.1558**	0.1535	0.1529	0.1525	0.1574	0.0247
120	0.1596	**0.1500**	0.1478	0.1460	0.1461	0.1510	0.0236

Table 5 Edge vector correlations (PCI) for the Hedgehog Method with repeated crankshaft rotations at various sampling rates as a function of the number n of edges. The mean PCI values (and their standard deviation) between two random sequences of n unit vectors without any closure condition are given in the columns labeled "random" to provide a benchmark for comparison. For each value of n, the PCI that comes closest to this random mean correlation is indicated in bold font

Edges	Crankshaft rotations					Random	
$n = 12k$	n	$2n$	$3n$	$4n$	$5n$	mean	std dev
12	0.3473	**0.3285**	0.3269	0.3295	0.3267	0.3280	0.0828
24	**0.2671**	0.2626	0.2611	0.2636	0.2624	0.2695	0.0605
36	**0.2318**	0.2261	0.2280	0.2252	0.2253	0.2352	0.0484
48	**0.2117**	0.2070	0.2065	0.2046	0.2052	0.2127	0.0413
60	**0.1943**	0.1889	0.1903	0.1875	0.1883	0.1972	0.0360
72	**0.1819**	0.1781	0.1766	0.1776	0.1760	0.1836	0.0319
84	**0.1701**	0.1688	0.1676	0.1673	0.1665	0.1732	0.0289
96	**0.1613**	0.1606	0.1588	0.1597	0.1593	0.1650	0.0269
108	**0.1543**	0.1534	0.1540	0.1522	0.1518	0.1574	0.0247
120	**0.1479**	0.1478	0.1476	0.1463	0.1473	0.1510	0.0236

sufficient for both the Hedgehog Method and the Triangle Method using repeated polygonal folds. In comparison, the data in Table 7 offers much less consistent results for the Triangle Method with repeated crankshaft rotations. Nevertheless, a conservative sampling rate of $3n$ crankshaft rotations seems adequate in this situation. In fact, in all of these cases, a sampling rate of n iterations appears to be sufficient to achieve PCI values within one standard deviation of the random distribution without the closure condition.

Table 6 Edge vector correlations (PCI) for the Triangle Method with repeated polygonal folds at various sampling rates as a function of the number n of edges. The mean PCI values (and their standard deviation) between two random sequences of n unit vectors without any closure condition are given in the columns labeled "random" to provide a benchmark for comparison. For each value of n, the PCI that comes closest to this random mean correlation is indicated in bold font

Edges	Polygonal folds					Random	
$n = 12k$	n	$2n$	$3n$	$4n$	$5n$	mean	std dev
12	**0.3092**	0.3078	0.3054	0.3066	0.3036	0.3280	0.0828
24	**0.2554**	0.2495	0.2531	0.2483	0.2532	0.2695	0.0605
36	**0.2199**	0.2174	0.2174	0.2171	0.2170	0.2352	0.0484
48	**0.1978**	0.1965	0.1968	0.1969	0.1970	0.2127	0.0413
60	**0.1811**	0.1800	0.1786	0.1808	0.1800	0.1972	0.0360
72	**0.1683**	0.1681	0.1685	0.1690	0.1687	0.1836	0.0319
84	**0.1585**	0.1590	0.1587	0.1576	0.1578	0.1732	0.0289
96	**0.1511**	0.1504	0.1495	0.1505	0.1498	0.1650	0.0269
108	**0.1444**	0.1425	0.1437	0.1442	0.1441	0.1574	0.0247
120	**0.1376**	0.1372	0.1367	0.1367	0.1381	0.1510	0.0236

Table 7 Edge vector correlations (PCI) for the Triangle Method with repeated crankshaft rotations at various sampling rates as a function of the number n of edges. The mean PCI values (and their standard deviation) between two random sequences of n unit vectors without any closure condition are given in the columns labeled "random" to provide a benchmark for comparison. For each value of n, the PCI that comes closest to this random mean correlation is indicated in bold font

Edges	Crankshaft rotations					Random	
$n = 12k$	n	$2n$	$3n$	$4n$	$5n$	mean	std dev
12	0.3680	0.3652	0.3669	**0.3625**	0.3649	0.3280	0.0828
24	0.2842	0.2787	0.2761	**0.2755**	0.2794	0.2695	0.0605
36	0.2439	0.2383	0.2427	0.2381	**0.2371**	0.2352	0.0484
48	0.2150	0.2139	0.2143	**0.2124**	0.2132	0.2127	0.0413
60	0.1986	**0.1965**	0.1968	0.1955	0.1956	0.1972	0.0360
72	**0.1826**	0.1828	0.1840	0.1826	0.1828	0.1836	0.0319
84	0.1744	**0.1733**	0.1721	0.1721	0.1707	0.1732	0.0289
96	0.1661	**0.1628**	0.1632	0.1627	0.1642	0.1650	0.0269
108	0.1581	**0.1565**	0.1557	0.1572	0.1562	0.1574	0.0247
120	**0.1511**	0.1502	0.1504	0.1492	0.1494	0.1510	0.0236

4.3 Computation Time Comparison

Since computational time is often an important factor in determining the size of the data set employed in a study, we have estimated the CPU time required by the *Mathematica* implementation of each of the algorithms discussed above. More precisely, we approximated the average CPU time (in seconds) associated with each iteration of these algorithms by calling on the `TimeUsed` command. Of course, it must be noted that the data reported by this command depends on many factors, including memory caching, "memoization," and other internal optimizations which depend on the precise state of the *Mathematica* session at the time the computations are performed [73]. To avoid many of these issues, we restarted

Table 8 Iteration time (in seconds) for the Polygonal Folding Method using various sampling rates as a function of the number n of edges

Number of edges	Number of polygonal folds				
$n = 12k$	n	$2n$	$3n$	$4n$	$5n$
12	0.0176	0.0375	0.0620	0.0889	0.0912
24	0.0857	0.1588	0.2290	0.3375	0.3380
36	0.1888	0.3610	0.5113	0.7283	0.7769
48	0.3316	0.6308	0.9686	1.2679	1.2288
60	0.5431	1.0081	1.4474	1.8528	2.1093
72	0.7907	1.3783	2.0695	2.7889	3.1104
84	1.0573	1.8293	2.6880	3.7830	4.4460
96	1.3632	2.3366	3.4377	4.7942	5.9450
108	1.7521	3.0207	4.6758	5.7636	7.5260
120	2.1440	3.7336	5.8153	7.1627	9.3135

Table 9 Iteration time (in seconds) for the Crankshaft Rotation Method using various sampling rates as a function of the number n of edges

Number of edges	Number of crankshaft folds				
$n = 12k$	n	$2n$	$3n$	$4n$	$5n$
12	0.0060	0.0081	0.0101	0.0122	0.0154
24	0.0140	0.0184	0.0223	0.0263	0.0349
36	0.0244	0.0323	0.0368	0.0429	0.0572
48	0.0372	0.0506	0.0537	0.0643	0.0835
60	0.0538	0.0704	0.0760	0.0895	0.1122
72	0.0735	0.0919	0.1048	0.1135	0.1369
84	0.0958	0.1168	0.1310	0.1457	0.1796
96	0.1214	0.1405	0.1635	0.1801	0.2230
108	0.1476	0.1745	0.1976	0.2138	0.2457
120	0.1816	0.2096	0.2316	0.2524	0.3031

the kernel before running each implementation. Thus, all other factors being equal, the consequences of timing our algorithms in *Mathematica* should have been kept to a minimum and distributed evenly among all four methods. Nevertheless, we must be careful not to overreach as we draw conclusions based on data that may be influenced by these issues. In particular, although the absolute times are not important in and of themselves, we assert that their relative size do provide a useful measure of the intrinsic complexity of the algorithms in question.

4.3.1 Computations

As in Sect. 4.2, we constructed two sample sets, each consisting of 1 000 polygons with $12k$ edges for each $k = 1, \ldots, 10$, by starting with a standard regular polygon and repeatedly applying polygonal folds or crankshaft rotations, respectively. In each case, the total CPU time was measured after n, $2n$, $3n$, $4n$, and $5n$ iterations. The average times are shown in Tables 8 and 9.

Two additional sample sets, once again consisting of 1 000 polygons with $12k$ edges for each $k = 1, \ldots, 10$, were generated by the Hedgehog and the Triangle Methods, respectively. These were subjected to a sequence of repeated crankshaft rotations and, as above, the total

Table 10 Iteration time (in seconds) for the Hedgehog Method with repeated crankshaft rotations at various sampling rates as a function of the number n of edges

Number of edges	Number of crankshaft folds				
$n = 12k$	n	$2n$	$3n$	$4n$	$5n$
12	0.0100	0.0127	0.0141	0.0237	0.0217
24	0.0228	0.0296	0.0365	0.0523	0.0503
36	0.0404	0.0509	0.0649	0.0759	0.0837
48	0.0619	0.0776	0.0986	0.0940	0.1304
60	0.0876	0.1082	0.1110	0.1254	0.1577
72	0.1108	0.1382	0.1465	0.1542	0.2054
84	0.1288	0.1455	0.1554	0.1922	0.2279
96	0.1615	0.1744	0.2060	0.2339	0.2692
108	0.1605	0.2089	0.2539	0.2935	0.3150
120	0.1857	0.2532	0.3022	0.3443	0.3602

Table 11 Iteration time (in seconds) for the Triangle Method with repeated crankshaft rotations at various sampling rates as a function of the number n of edges

Number of edges	Number of crankshaft folds				
$n = 12k$	n	$2n$	$3n$	$4n$	$5n$
12	0.0160	0.0202	0.0203	0.0271	0.0287
24	0.0358	0.0443	0.0503	0.0569	0.0594
36	0.0625	0.0751	0.0854	0.0951	0.0912
48	0.0949	0.1098	0.1251	0.1377	0.1394
60	0.1326	0.1509	0.1711	0.1863	0.1835
72	0.1760	0.1951	0.2173	0.2377	0.2234
84	0.1819	0.2049	0.2335	0.2599	0.2608
96	0.2267	0.2509	0.2798	0.3112	0.3075
108	0.2713	0.3030	0.3369	0.3674	0.3588
120	0.3155	0.3511	0.3968	0.4200	0.4208

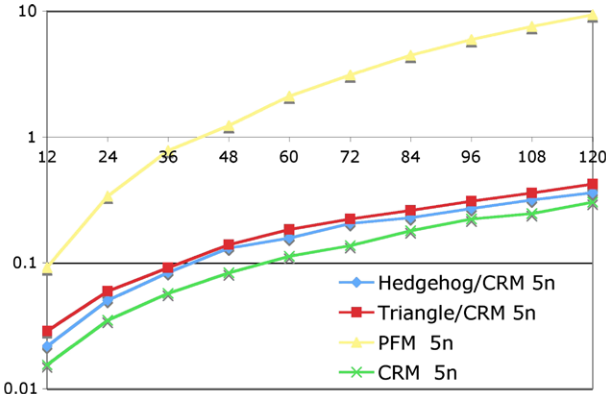


Fig. 7 (Color online) A comparison of the average CPU time (in seconds) required for $5n$ iterations of the Polygonal Folding, Crankshaft Rotation, Hedgehog, and Triangle Methods as a function of the number n of edges. Both the Hedgehog and Triangle Methods were implemented using repeated crankshaft rotations

CPU time was measured after n, $2n$, $3n$, $4n$, and $5n$ iterations. The average times are shown in Tables 10 and 11. Note that these values include the time necessary to generate the initial embedded configuration, including any permutations of the initial edge vectors, as discussed in Sect. 4.1, in addition to the time required to perform the appropriate number of crankshaft rotations.

Figure 7 presents a graphical comparison of all of four computational times (as a function of the number n of edges) after $5n$ iterations.

4.3.2 Analysis

There are three important points to be observed from the data presented in these tables. First of all, we note that the implementation of the Polygonal Folding Method is dramatically slower than that of the Crankshaft Rotation Method, as illustrated in Tables 8 and 9. We surmise that the difference is a direct consequence of the number of computations required by each basic step. More specifically, each crankshaft rotation modifies only the two edge vectors selected, while each polygonal fold requires an average of roughly $\frac{n}{3}$ vertex (or edge) modifications.

Secondly, our data in Tables 10 and 11 shows that the iteration times for the Triangle Method are consistently larger than those for the Hedgehog Method. Since the only difference in the implementation of these algorithms is the initial data generation, this seems to imply that the principal factor contributing to the Triangle Method's slower performance is precisely the generation and permutation of the initial edge vectors. As expected, this overhead grows in significance as the number of edges increases.

Finally, we note that average iteration times for the Hedgehog and the Crankshaft Rotation Methods in Tables 9 and 10 are generally comparable to one another. In fact, they do not seem to differ by more than a few milliseconds per iteration. This certainly adds credence to our hypothesis above, concerning the slower behavior of the Triangle Method, since it does not appear that the overhead involved in the Hedgehog Method's initialization is significantly hampering its overall performance.

4.3.3 Conclusions

We believe that one may safely draw several conclusions from these comparisons. To begin with, the Polygonal Folding Method is substantially slower than both the Crankshaft Rotation and the Hedgehog Methods, most likely due of its computationally intensive steps. Similarly, the initial data preparation time required by the Triangle Method is a slight hindrance to the implementation of this algorithm, especially when compared to the Crankshaft Rotation and the Hedgehog Methods. Finally, it seems that the Hedgehog Method performs about as well as the Crankshaft Rotation Method, even given the overhead associated to its initial data preparation. Taking into consideration its small sampling rate, as determined in Sect. 4.2, one begins to see the benefits of implementing the classical Hedgehog Method with repeated crankshaft rotations as the preferred algorithm for generating random polygons.

4.4 Edge Vector Distribution on the Unit Sphere

We now turn our attention to the distribution of edge vectors on the unit sphere. We wish to determine how randomly our four algorithms select edge vectors, when compared to an unconstrained set of random unit vectors. Taking into consideration the closure requirement, one may expect that our random polygons experience a biasing influence on the distribution of their edges on the sphere. To test the degree of uniformity of the edge vector distribution of a polygon, we partition the sphere into 625 disjoint regions. We then count the number of edge vectors in each of these regions. If the number of vectors is large, and they are uniformly distributed, one would expect that their distribution among these regions would be roughly proportional to the distribution of area among these regions. We shall use the χ^2 Goodness of Fit Test [38] to measure the degree of consistency between two such distributions.

4.4.1 Computations

Each of the four algorithms was used to produce a collection of random $12k$-edge equilateral polygons for each integer $k = 1, \ldots, 10$. As in Sects. 4.2 and 4.3 above, five distinct sampling rates ranging from n to $5n$ were used. The results for each algorithm and each sampling rate were grouped into 100 families, each consisting of 1 000 random equilateral polygons. For each polygon, we determined the distribution of the edge vectors among the 625 regions of the sphere, and, using the χ^2 Goodness of Fit Test, we estimated the consistency of the data against the corresponding distribution of area among the regions. For each family of polygons, we computed the proportion of polygons that satisfied the χ^2 test with a level of consistency of 95%. These proportions were averaged together and recorded in Tables 12, 13, 14, and 15. A comparison of the relative effectiveness of each of the algorithms after $5n$ iterations is found in Table 16 and Fig. 8.

In order to establish a benchmark for spherical uniformity, 100 families, each consisting of 1 000 sample sets of $12k$ random unit vectors (without any closure condition), were generated for each integer $k = 1, \ldots, 10$. For each one of these sample sets, the distribution of vectors among the 625 regions of the sphere was evaluated for consistency by the χ^2 Goodness of Fit Test, as above, and the proportion of sets in each family satisfying this test with a 95% level of consistency was computed. Finally, these proportions were averaged together to produce our target benchmarks. These values appear in the column labeled “random” in Tables 12, 13, 14, and 15. Furthermore, Table 16 and Fig. 8 provide a direct comparison between this random measure of spherical uniformity and the corresponding values after $5n$ iterations for each of our four algorithms.

4.4.2 Analysis

Because we know that the distribution of individual edge vectors of a polygon is constrained by the closure condition, one might expect that the algorithms will fall short of the benchmark values to some systematic extent. This effect, however, does not seem to be apparent in our data. To the contrary, the effects of closure and knotting appear to increase the degree of uniformity. The data suggests that, with increasing numbers of edges, the difference in the degree of uniformity as compared to that of random vectors may be relatively constant, as seen in Fig. 8.

4.4.3 Conclusions

Each of the algorithms appears to produce a level of spherical uniformity comparable to that of a set of random vectors. In fact, as Fig. 8 shows, there seems to be a systematic tendency to achieve greater uniformity as a consequence, perhaps, of the closure of the polygons. The data also suggests that the potential increase in uniformity is nearly constant across the number of edges.

Table 12 Spherical uniformity of edge vector distributions (as a percentage) produced by the Polygonal Folding Method using various sampling rates. The mean values in the column labeled "random," corresponding to a collection of random unit vectors without any closure condition, provide a benchmark for comparison

Number of edges	Number of crankshaft rotations					Random
$n = 12k$	n	$2n$	$3n$	$4n$	$5n$	
12	84.19	83.32	82.38	82.22	82.21	78.10
24	78.55	77.55	77.77	78.16	78.58	75.94
36	76.64	77.01	77.27	76.39	76.84	75.08
48	74.74	78.50	78.49	79.45	78.18	76.57
60	67.56	78.59	80.17	80.82	81.07	78.17
72	61.78	79.32	81.83	81.43	81.23	79.53
84	55.14	78.91	81.87	82.48	81.00	81.10
96	49.24	79.05	82.98	82.96	83.34	82.40
108	43.63	79.73	83.39	83.85	84.49	83.50
120	38.54	78.90	84.27	85.20	85.08	84.30

Table 13 Spherical uniformity of edge vector distributions (as a percentage) produced by the Crankshaft Rotation Method using various sampling rates. The mean values in the column labeled "random," corresponding to a collection of random unit vectors without any closure condition, provide a benchmark for comparison

Number of edges	Number of crankshaft rotations					Random
$n = 12k$	n	$2n$	$3n$	$4n$	$5n$	
12	91.41	83.28	79.83	79.52	78.77	78.10
24	90.12	81.84	78.56	77.22	76.70	75.94
36	91.57	83.61	78.12	76.03	75.54	75.08
48	93.60	85.71	81.00	78.47	78.12	76.57
60	95.56	89.74	84.54	83.44	82.75	78.17
72	95.32	88.88	84.49	81.27	80.45	79.53
84	95.58	90.42	85.50	82.68	81.97	81.10
96	95.92	91.53	86.67	83.31	83.60	82.40
108	95.31	92.47	87.49	85.31	83.09	83.50
120	95.25	93.26	88.75	86.67	85.73	84.30

4.5 Local Curvature and Torsion Distributions

The four methods discussed here have been applied throughout the literature to make estimations of many potentially interesting measures of the shape of a polygon, including the radius of gyration, the average crossing number, the diameter, the miniball radius, the box and skinny box dimensions (as well as the associated surface area and volume), the volume and surface area of the convex hull of the polygon, the inertial asphericity, and the enveloping ellipsoid asphericity [11, 17, 18, 20, 21, 39, 59, 62, 63, 66, 67]. Another application of particular importance for this study is the estimation of the total curvature and total torsion of the polygon [60].

Generalizing the notions of curvature and torsion for smooth curves [19], the local curvature at each vertex of a polygon is defined to be the turning angle at that vertex, while the local torsion at each edge of a polygon is defined to be the dihedral angle between the

Table 14 Spherical uniformity of edge vector distributions (as a percentage) produced by the Hedgehog Method with repeated crankshaft rotations. The mean values in the column labeled "random," corresponding to a collection of random unit vectors without any closure condition, provide a benchmark for comparison

Number of edges	Number of crankshaft rotations					Random
$n = 12k$	n	$2n$	$3n$	$4n$	$5n$	
12	78.24	76.85	76.54	77.10	76.44	78.10
24	75.25	77.24	77.67	77.55	77.43	75.94
36	77.61	76.52	76.23	75.56	75.02	75.08
48	76.40	76.91	77.60	78.18	76.83	76.57
60	74.63	79.40	78.64	79.28	78.57	78.17
72	75.37	80.18	80.23	79.53	80.62	79.53
84	75.99	81.35	81.61	82.51	81.04	81.10
96	87.28	87.15	87.14	87.38	87.07	82.40
108	87.97	87.48	87.91	87.78	88.36	83.50
120	88.70	88.89	88.59	88.47	88.18	84.30

Table 15 Spherical uniformity of edge vector distributions (as a percentage) produced by the Triangle Method with repeated crankshaft rotations. The mean values in the column labeled "random," corresponding to a collection of random unit vectors without any closure condition, provide a benchmark for comparison

Number of edges	Number of crankshaft rotations					Random
$n = 12k$	n	$2n$	$3n$	$4n$	$5n$	
12	78.50	75.95	77.00	77.95	76.45	78.10
24	75.50	77.00	77.40	76.35	76.45	75.94
36	72.50	80.30	79.50	80.50	79.95	75.08
48	76.00	82.60	82.10	82.30	83.10	76.57
60	78.30	82.60	84.10	83.30	83.85	78.17
72	79.40	85.30	84.85	84.25	84.20	79.53
84	80.60	86.65	85.25	85.75	85.55	81.10
96	87.13	86.99	87.55	86.51	86.50	82.40
108	87.59	87.76	87.90	87.91	87.66	83.50
120	88.68	88.55	88.46	88.39	88.36	84.30

two planes determined by the edge and each of its adjacent edges. Consequently, the total curvature is defined to be the sum of the turning angles, and the total torsion is defined to be the sum of the dihedral angles. Evidence has shown that the effects of curvature and torsion may be reflected in the physical properties of polymers [60].

The average total curvature and the average total torsion for a collection of random equilateral polygons have been shown to differ from the expected values for a random collection of vectors [30, 60]. These differences indicate a certain degree of internal correlation among the edge vectors of a polygon, arising as a consequence of the closure condition of the polygon. In fact, the average total curvature of an equilateral polygon contains a bias of $+\frac{3\pi}{8}$, while the average total torsion of an equilateral polygon contains a bias of $-\frac{3\pi}{8}$ [30, 60].

We shall now consider the distributions of local curvature and local torsion angles and compare these to the distributions arising from a random sequence of edges without the

Table 16 Spherical uniformity of edge vector distributions (as a percentage) produced after $5n$ iterations of each algorithm. The mean values in the column labeled "random," corresponding to a collection of random unit vectors without any closure condition, provide a benchmark for comparison

Edges	PFM	CRM	Hedgehog	Triangle	Random
12	82.22	79.52	77.10	77.95	78.10
24	78.16	77.22	77.55	76.35	75.94
36	76.39	76.03	75.56	80.50	75.08
48	79.45	78.47	78.18	82.30	76.57
60	80.82	83.44	79.28	83.30	78.17
72	81.43	81.27	79.53	84.25	79.53
84	82.48	82.68	82.51	85.75	81.10
96	82.96	83.31	87.38	86.51	82.40
108	83.85	85.31	87.78	87.91	83.50
120	84.27	86.67	88.47	88.39	84.30

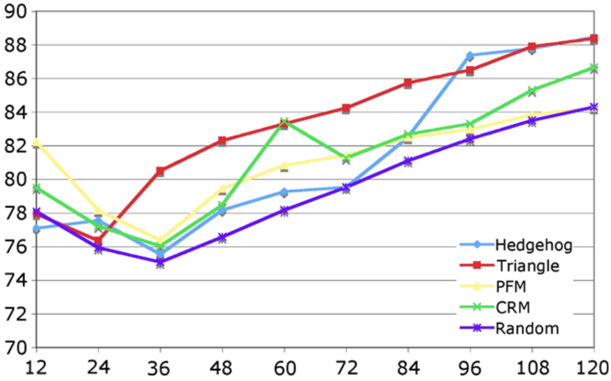


Fig. 8 (Color online) A comparison of the spherical uniformity of edge vector distributions (as a percentage) after $5n$ iterations as a function of the number n of edges. The curve labeled "random," corresponding to a collection of random unit vectors without any closure condition, provide a benchmark for comparison

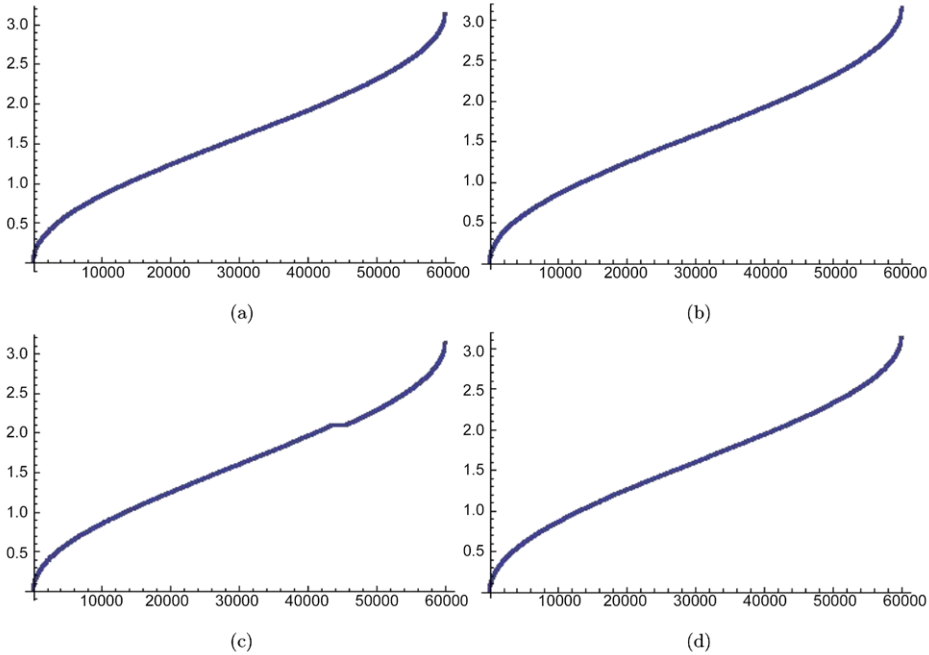
closure constraint. We will observe that it may be necessary to employ sufficiently many polygonal folds or crankshaft rotations in order to eliminate curvature and torsion relationships arising in the initial data produced by the Hedgehog or Triangle Methods.

4.5.1 *Computations*

We collected and analyzed 10 000 60-edge polygons, grouped in sample sets of 1 000 polygons each, using the *Mathematica* implementations of each of our four algorithms, employing a sampling rate of $3n = 180$ polygonal folds (for the PFM) or crankshaft rotations (for the CRM, the Hedgehog Method, and the Triangle Method). In addition, we analyzed the data collected from the Triangle Method both before and after the application of crankshaft rotations. For each sample set, we computed the local curvature and torsion angles, and sorted these in increasing order. The Kolmogorov-Smirnov Goodness of Fit Test [8] was then employed to quantify the similarity of the resulting distributions, thus providing a measure of the likelihood that they arose from the same probability distribution function.

4.5.2 *Analysis*

Figure 9 shows a graphical representation of the average distribution of local curvature for (a) a benchmark collection of 60 000 random unit vectors without the closure condition,



(a)

(b)

(c)

(d)

Fig. 9 Curvature distributions for (**a**) a random collection, (**b**) the Crankshaft Rotation Method, (**c**) the Triangle Method without crankshaft rotations, and (**d**) the Triangle Method with crankshaft rotations. Note the anomalous flat segment at $\kappa = 2\pi/3$ in the middle of the graph in (**c**). This plateau is a consequence of the small but positive probability that two edges from the same equilateral triangle end up in adjacent positions in the polygon after the initial permutation of edges, and disappears after the application of an adequate number of crankshaft rotations

(b) the Crankshaft Rotation Method, (c) the Triangle Method prior to any crankshaft rotations, and (d) the Triangle Method after repeated crankshaft rotations. In the case of the Triangle Method without crankshaft rotations, we note an anomalous flat segment at $\kappa = \frac{2\pi}{3}$ in the middle of the graph in Fig. 9(c). This plateau is a consequence of the small but positive probability that two edges from the same equilateral triangle end up in adjacent positions in the polygon after the initial permutation of edges. Therefore, for applications in which local geometry might be important, it would appear advisable to employ polygonal folds or crankshaft rotations to eliminate these relationships. Nevertheless, according to the Kolmogorov-Smirnov test, there is a strong agreement between the benchmark curvature distribution in Fig. 9(a) and the curvature distributions for each of our four methods. For instance, in the case of the curvature distributions for the Triangle method prior to and after the application of crankshaft rotations, the statistic D_{mn} takes on values of 0.4623 and 0.1036, with maximum distances of 0.00642 and 0.00144, respectively. Since these values of D_{mn} are smaller than the critical value of 1.35, we can conclude at the 5% significance level that the curvatures are likely to arise from the same probability distribution function. It turns out that the PFM, CRM, and Hedgehog Methods also generate polygons that, with respect to local curvature, agree with the expected distributions.

Figure 10 shows a graphical representation of the average distribution of torsion for (a) a benchmark collection of 60 000 random unit vectors without the closure condition, (b) the Crankshaft Rotation Method, (c) the Triangle Method prior to any crankshaft rotations, and (d) the Triangle Method after repeated crankshaft rotations. According to the Kolmogorov-

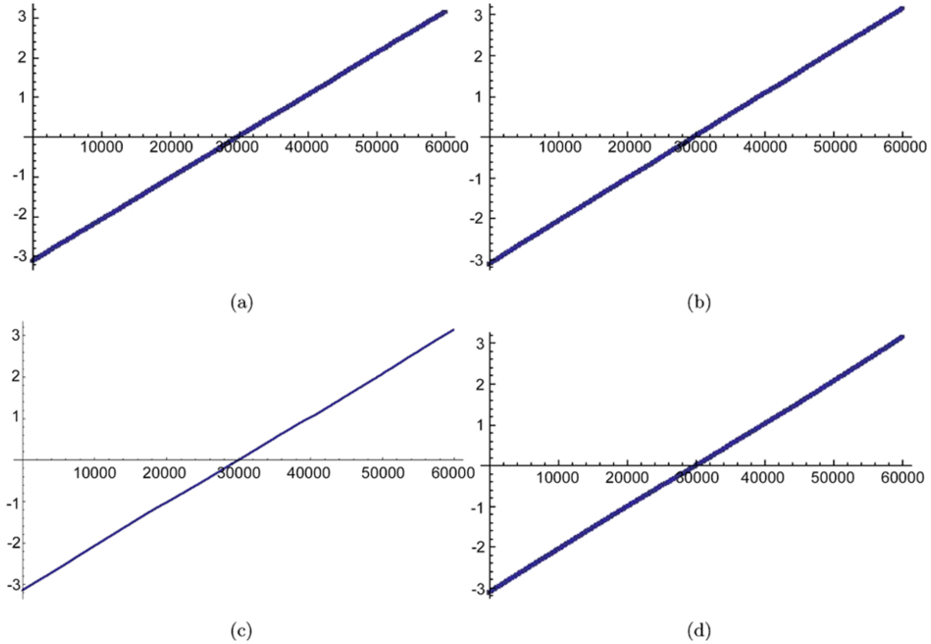


Fig. 10 Torsion distributions for (**a**) a random collection, (**b**) the Crankshaft Rotation Method, (**c**) the Triangle Method without crankshaft rotations, and (**d**) the Triangle Method with crankshaft rotations. As in the case of curvature (see Fig. 9), there is a small (essentially indiscernible) flat segment corresponding to a dihedral angle of $\tau = 0$ in the middle of the graph in (**c**); this is due to the very small but positive probability that three adjacent edges belong to the same equilateral triangle. As before, introducing polygonal folds or crankshaft rotations quickly eliminates this problem

Smirnov test, the local torsion produced by the various algorithms has a similarly robust distribution. Once again, the exceptional case here is the distribution produced by the Triangle Method prior to crankshaft rotations. In this case, there is a small but positive probability that three adjacent edges belong to the same equilateral triangle. Such instances give a small (essentially indiscernible to the naked eye) flat segment corresponding to a dihedral angle of $\tau = 0$. As before, introducing polygonal folds and crankshaft rotations quickly eliminates this problem.

4.5.3 Conclusions

Our study suggests that all four methods for generating random equilateral polygons yield consistent distributions of local curvature and local torsion angles. The only exceptions appear to come from the Triangle Method prior to the application of crankshaft rotations, where an anomalous abundance of curvature angles of $\kappa = \frac{2\pi}{3}$ and torsion angles of $\tau = 0$ arise from adjacent edge vectors originating from the same equilateral triangles. In all cases, these anomalous angles disappeared after the repeated application of crankshaft rotations.

4.6 Knot Probability Distributions

The configurations generated by our four algorithms are often used to estimate a wide range of average properties of polygonal knots. In these applications, the extent to which the sam-

Table 17 The number of distinct HOMFLY polynomials produced by each algorithm. Since the number of distinct knots sharing the same polynomial is small, this invariant is an suitable surrogate for knot type

Method	Sample size	Distinct HOMFLY
PFM	100 000 000	2 219
CRM	10 000 000	6 110
Hedgehog	10 000 000	1 111
Triangle	10 000 000	3 505

ple configurations represent the statistics of the entire population of knots is a critical factor in evaluating any implications drawn from these estimates. With this in mind, the final benchmark with which we test the quality of our data is the probability distribution function of the various knot types, that is, the proportion of equilateral knot space corresponding to polygons of a specific knot type [12, 13, 46–48].

While a rigorous determination of the knot probability distribution function has not yet been accomplished, we do have some theorems and estimates concerning the asymptotic probability of the unknot. For an excellent review of what is known about the asymptotic distributions for unknots and other individual knot types in both the lattice and continuum settings, we refer to the excellent review article of Orlandini and Whittington [58]. They discuss both theoretical bounds and numerical estimates of the exponential decay as a function of the number of edges. Nevertheless, we encounter the problem of not knowing exactly the limiting decay or where this asymptotic behavior actually begins, and thus risk that it lies beyond the range of interest. Additionally, even though there is only a finite number of knot types for any fixed number of edges, there is no good estimate of just how large this number might be as a function of the number of edges.

Due to the difficulty of classifying the possible knot types that might be encountered [32, 33], researchers have employed a variety of strategies to give indirect estimates of the number of distinct knot types and their probabilities as a function of the number of edges. For example, the Alexander, Jones, or HOMFLY polynomial invariants may be used as a surrogate for knot type, although other options are also worthy of consideration. In this study, we shall use the Ewing-Millett algorithm to calculate the HOMFLY knot invariant [24, 27]. Although not a fully faithful representation of knot type, it is somewhat stronger than many of the other options, especially for our purposes. In particular, there are relatively few classes of duplications to undermine the robustness of the statistical analysis using this invariant [32].

4.6.1 Computations

In this study, we will use the Kolmogorov-Smirnov test to assess whether or not the data developed from the various methods is a consequence of the same probability distribution function, thereby testing the relative quality of the generation algorithms. As before, we will focus our analysis on 60-edge polygonal knots, for which we have a benchmark sample of 100 million knots previously generated by the Polygonal Folding Method [47, 48]. We will use this benchmark to compare a collection of 10 million polygons produced by each of the other three algorithms studied here. In all four cases, the samples were constructed by the *Fortran 77* implementation of our algorithms using a sampling rate of $3n = 180$ polygonal folds (for the PFM) or crankshaft rotations (for the CRM, the Hedgehog Method, and the Triangle Method) per sample.

Using the Ewing-Millett program, we determined the HOMFLY polynomial of each polygon generated. Table 17 gives a summary of the number of distinct polynomials observed in each sample. We note that the Crankshaft Rotation Method produces many more

Table 18 Results of the Kolmogorov-Smirnov test comparing the distribution of knot types produced by PFM with those produced by the other algorithms. Since the values of the D_{mn} statistic is less than the critical value of 1.35 for this data, the data from the four algorithms appear to be consistent with the knot probability distribution function for equilateral polygonal knots

Comparison	D_{mn}	Max distance
PFM v. CRM	0.030	0.000 7
PFM v. Hedgehog	0.184	0.005 8
PFM v. Triangle	0.495	0.011 0

polynomials and, therefore, identifies many more topological knot types than any of the other algorithms. To test whether or not the algorithms sampled data from the same knot probability distribution, the proportions of the resulting polynomials were sorted in decreasing order and their ranking functions were compared using the Kolmogorov-Smirnov test. Since the Hedgehog Method produced the smallest number of polynomials (only 1 111 distinct polynomials), we restricted our comparison to only the first 1 000 more likely polynomials from each collection. In particular, as the tails of the distributions represent polynomials arising only once, the fact that they may give slightly different instances should not adversely distort this comparison.

4.6.2 Analysis

Table 18 presents the results of the Kolmogorov-Smirnov test comparing the benchmark PFM data to that of the other three algorithms. The values in the “max distance” column of the table give the largest differences between corresponding values in the two probability distribution functions being compared. The observed values suggest that the two distributions are quite similar. Indeed, since each of the Kolmogorov-Smirnov statistic values D_{mn} is less than the critical value of 1.35 for this data, we may conclude that the four algorithms give data consistent with the knot probability distribution function for equilateral polygonal knots.

4.6.3 Conclusions

Our analysis above leads to a couple of observations. First of all, by restricting our attention to the $1,000$ most frequently observed knot types, we find that the four algorithms do appear to be sampling the same knot probability function, according to the Kolmogorov-Smirnov test. Nevertheless, the apparently large differences in the number of distinct polynomials observed in each of the data sets suggest that one may wish to look closely at the objectives of the simulation in making a choice of generation method. In particular, there may be underlying differences making one approach more appropriate than another. For instance, according to the implementations reported here, the use of CRM produced superior results than any of the other sampling methods, including the Hedgehog or Triangle Methods. We recommend further careful study of the algorithms to confirm these observations.

5 Final Remarks

We have described four popular methods to generate random equilateral polygons in three-space: the Polygonal Folding Method, the Crankshaft Rotation Method, the Hedgehog

Method, and the Triangle Method. Here, we give the first rigorous proof that the Crankshaft Rotation Method is ergodic (see Theorem 2). We also show that this provides a fast, attractive alternative to the Polygonal Folding Method, which was already known to be ergodic.

Both the Hedgehog and Triangle Methods have modest constraints, in that they require about $1.56n$ and $0.77n$ initial edge transpositions, respectively, in order to produce polygons with distinct vertices (see Fig. 6). This overhead does not appear to be significantly hampering the performance of either of these algorithms, especially when compared to the drastically slower behavior of the Polygonal Folding Method (see Fig. 7). With regard to sampling rate, we propose that $2n$ polygonal folds or crankshaft rotations are typically sufficient to insure independence of samples, and that, in some cases, one could employ as few as n elementary transformations (see Tables 2 to 7). All four methods appear to give distributions of edge vectors in the unit sphere comparable to that of a random collection of vectors with no closure condition, especially when a sufficient number of repeated polygonal folds or crankshaft rotations is applied (see Fig. 8). Similarly, all four methods produce consistent distributions of local curvature and local torsion, according to the Kolmogorov-Smirnov goodness of fit test.

In our simulations, the Crankshaft Rotation Method appears to be the strongest, in the sense that it may access the largest number of distinct topological knot types (see Table 17). Even though one would expect the Hedgehog or Triangle Methods to do at least as well as the CRM, the range of knots uncovered by these two methods seems to be somehow hampered by the intrinsic symmetries in their initial configurations. Further research may be needed to determine if this is, in fact, the case. Despite this concern, we have shown, using the Kolmogorov-Smirnov test, that all four methods appear to sample the same knot probability distribution function, at least with respect to the most prevalent 1,000 knot types (see Table 18).

As discussed in the introduction, these strategies are well adapted to studies in which the consequences of excluded volume are neglected. Are they useful when one must take such interactions into consideration? We believe that they answer is yes, when used in conjunction with novel statistical methods such as multiple Markov chains or parallel tempering [25, 28, 42, 44, 45, 65, 68] due to the relative speed achievable in implementing, for example, the crankshaft rotation algorithm. Similarly, these algorithms are useful, again due to their relative speed, when one wishes to sample the subsets consisting of a conformations having a specified geometric property, e.g. a range of diameters, or a specific knot type. In such cases, one employs the algorithms to sample the entire space and applies a filter, such as the HOMFLY polynomial in the case of a selected knot type, to select those lying in the desired subspace. While this approach is computationally intensive, it has been applied to study the effects of confinement [48] or the spatial properties of a fixed knot type [52]. Ideally, one would like to know that these algorithms are ergodic when restricted to specific geometric or topological knot types, but this is unknown due to two fundamental problems. First, one does not know that the sets of specific knot types are connected, even in the case of the unknot. Second, in the event that they are not connected, one does not know how to ensure that the individual components are properly sampled. Each of these issues remains a problem even in the case of the cubic lattice when one fixes the number of edges. On the other hand, there is a method, the BFACF algorithm [58], which does preserve the knot type at the cost of allowing the number of edges to change. Properly sampling the set of knots of a fixed type and fixed number of edges, with or without taking into consideration the consequences of excluded volume, remains a objective for further research.

Acknowledgements The authors wish thank Morwen Thistlethwaite for his assistance in extending the range of our simulations, as well as Michael Piatek, Eric Rawdon, and Andrzej Stasiak for their collaborations and discussions on simulations projects leading to the work we report here.

References

1. Amzallag, A., Vaillant, C., Jacob, J., Unser, M., Bednar, J., Kahn, J.D., Dubochet, J., Stasiak, A., Maddocks, J.H.: 3d reconstruction and comparison of shapes of DNA minicircles observed by cryo-electron microscopy. Nucleic Acids Res. **34**, e125-1-8 (2006)
2. Arsuaga, J., Vazquez, M., Trigueros, S., Sumners, D.W., Roca, J.: Knotting probability of DNA molecules confined in restricted volumes: DNA knotting in phage capsids. Proc. Natl. Acad. Sci. USA **99**, 5373–5377 (2002)
3. Arsuaga, J., Vazquez, M., McGuirk, P., Trigueros, S., Sumners, D.W., Roca, J.: DNA knots reveal a chiral organization of DNA in phage capsids. Proc. Natl. Acad. Sci. USA **102**(26), 9165–9169 (2005)
4. Calvo, J.A.: Geometric knot theory: the classification of spatial polygons with a small number of edges. Ph.D. thesis, University of California, Santa Barbara (1998)
5. Calvo, J.A., Millett, K.C.: Minimal edge piecewise linear knots. In: Ideal Knots, pp. 107–128. World Sci. Publishing, Singapore (1998)
6. Calvo, J.A., Millett, K.C.: Minimal edge piecewise linear knots. In: Ideal Knots. Ser. Knots Everything, vol. 19, pp. 107–128. World Sci. Publ., River Edge (1998)
7. Cerny, V.: A thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. J. Optim. Theory Appl. **45**, 51–51 (1985)
8. Chakravarti, I.M., Laha, R.G., Roy, J.: Handbook of Methods of Applied Statistics, vol. I. Wiley, New York (1967)
9. de Gennes, P.G.: Collapse of a polymer chain in poor solvents. J. Phys. Lett. **36**, 55–57 (1975)
10. de Gennes, P.G.: Scaling Concepts in Polymer Physics. Cornell University Press, Ithaca (1979)
11. Deguchi, T., Shimamura, M.K.: Topological effects on the average size of random knots. In: Physical Knots: Knotting, Linking, and Folding Geometric Objects in $\mathbb{R}^3$. Contemp. Math., vol. 304, pp. 93–114. Amer. Math. Soc., Providence (2002)
12. Deguchi, T., Tsurusaki, K.: A statistical study of random knotting using the Vassiliev invariants. J. Knot Theory Ramif. **3**(3), 321–353 (1994). Random knotting and linking (Vancouver, BC, 1993)
13. Deguchi, T., Tsurusaki, K.: Numerical application of knot invariants and universality of random knotting. In: Knot Theory, Warsaw, 1995. Banach Center Publ., vol. 42, pp. 77–85. Polish Acad. Sci., Warsaw (1998)
14. des Cloizeaux, J.: Ring polymers in solution: Topological effects. J. Phys. Lett. **42**, 433–436 (1981)
15. Deutsch, J.M.: Equilibrium size of large ring molecules. Phys. Rev. E **59**(3), 2539–2541 (1999)
16. Diao, Y., Ernst, C., Janse van Rensburg, E.J.: In search of a good polygonal knot energy. J. Knot Theory Ramif. **6**(5), 633–657 (1997)
17. Diao, Y., Ernst, C.: The average crossing number of Gaussian random walks and polygons. In: Physical and Numerical Models in Knot Theory. Series on Knots and Everything, vol. 36, pp. 275–292. World Sci. Publ., River Edge (2005)
18. Diao, Y., Dobay, A., Kusner, R.B., Millett, K.C., Stasiak, A.: The average crossing number of equilateral random polygons. J. Phys. A **36**(46), 11,561–11,574 (2003)
19. do Carmo, M.P.: Differential Geometry of Curves and Surfaces. Prentice-Hall Inc., Englewood Cliffs (1976). Translated from the Portuguese
20. Dobay, A., Sottas, P., Dubochet, J., Stasiak, A.: Predicting optimal lengths of random knots. Lett. Math. Phys. **55**(3), 239–247 (2001). Topological and geometrical methods (Dijon, 2000)
21. Dobay, A., Dubochet, J., Millett, K.C., Sottas, P., Stasiak, A.: Scaling behavior of random knots. Proc. Natl. Acad. Sci. USA **100**(10), 5611–5615 (2003)
22. Drube, E., Alim, K., Witz, G., Dietler, G., Frey, E.: Excluded volume effects on semiflexible ring polymers. Nano Lett. **10**, 1445–1449 (2010)
23. Ercolini, E., Valle, E., Adamcik, J., Witz, G., Metzler, R., De Los Rios, P., Roca, J., Dietler, G.: Fractal dimension and localization of DNA knots. Phys. Rev. Lett. **98** 058102 (2007)
24. Ewing, B., Millett, K.C.: Computational algorithms and the complexity of link polynomials. In: Progress in Knot Theory and Related Topics, pp. 51–68. Hermann, Paris (1997)
25. Ferrenberg, A.M., Swendsen, R.H.: Application of the Monte Carlo method to the lattice-gas model. Phys. Rev. Lett. **61**, 2635–2638 (1988)
26. Forsén, S. (ed.): Nobel Lectures, Chemistry 1971–1980. World Scientific, Singapore (1993)
27. Freyd, P., Yetter, D., Hoste, J., Lickorish, W.B.R., Millett, K.C., Ocneanu, A.: A new polynomial invariant of knots and links. Bull., New Ser., Am. Math. Soc. **12**(2), 239–246 (1985)
28. Geyer, C.J.: Markov chain Monte Carlo maximum likelihood. In: Computing Science and Statistics: Proc. 23rd Symp. on the Interface, pp. 156–163 (1991)
29. Grosberg, A.Y.: Critical exponents for random knots. Phys. Rev. Lett. **85**(18), 3858–3861 (2000)
30. Grosberg, A.Y.: Total curvature and total torsion of a freely jointed circular polymer with $n \gg 1$ segments. Macromolecules **41**(12), 4524–4527 (2008)

31. Haahr, M.: True random number service. http://www.random.org. Cited Nov 2007
32. Hoste, J.: The enumeration and classification of knots and links. In: Handbook of Knot Theory, pp. 209–232. Elsevier, Amsterdam (2005)
33. Hoste, J., Thistlethwaite, M., Weeks, J.: The first 1,701,936 knots. Math. Intell. **20**(4), 33–48 (1998)
34. Kapovich, M., Millson, J.J.: The symplectic geometry of polygons in Euclidean space. J. Differ. Geom. **44**, 479–513 (1996)
35. Kirpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
36. Klenin, K.V., Vologodskii, A.V., Anshelevich, V.V., Dykhne, A.M., Frank-Kamenetskii, M.D.: Effect of excluded volume on topological properties of circular DNA. J. Biomol. Struct. Dyn. **5**, 1173–1185 (1988)
37. Lal, M.: Monte Carlo computer simulation of chain molecules. I. Mol. Phys. **17**, 57–64 (1969)
38. Larsen, R.J., Marx, M.L.: An Introduction to Mathematical Statistics and Its Applications, 3rd edn. Prentice Hall, Upper Saddle River (2001)
39. Le Bret, M.: Monte Carlo computation of the supercoiling energy. The sedimentation constant, and the radius of gyration of unknotted and knotted circular DNA. Biopolymers **19**(3), 619–637 (1980)
40. Madras, N., Slade, G.: The Self-Avoiding Walk. Birkhäuser, Boston (1996)
41. Madras, N., Sokal, A.D.: The pivot algorithm: A highly efficient Monte Carlo method for the self-avoiding walk. J. Stat. Phys. **50**, 109–186 (1988)
42. Marenduzzo, D., Orlandini, E., Stasiak, A., Sumners, D.W., Tubiana, L., Micheletti, C.: Application of the Monte Carlo method to the lattice-gas model. Proc. Natl. Acad. Sci. USA **106**, 22269–22274 (2009)
43. Metropolis, N.C., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. **21**, 1087–1092 (1953)
44. Micheletti, C., Marenduzzo, D., Orlandini, E., Sumners, D.W.: Application of the Monte Carlo method to the lattice-gas model. J. Chem. Phys. **124**, 64903-1-10 (2006)
45. Micheletti, C., Marenduzzo, D., Orlandini, E., Sumners, D.W.: Application of the Monte Carlo method to the lattice-gas model. Biophys. J. **95**, 3591–3599 (2008)
46. Millett, K.C.: Knotting of regular polygons in 3-space. J. Knot Theory Ramif. **3**(3), 263–278 (1994). Random knotting and linking (Vancouver, BC, 1993)
47. Millett, K.C.: Monte Carlo explorations of polygonal knot spaces. In: Knots in Hellas '98 (Delphi). Ser. Knots Everything, vol. 24, pp. 306–334. World Sci. Publ., River Edge (2000)
48. Millett, K.C.: An investigation of equilateral knot spaces and ideal physical knot configurations. In: Physical Knots: Knotting, Linking, and Folding Geometric Objects in $\mathbb{R}^3$. Contemp. Math., vol. 304, pp. 77–91. Amer. Math. Soc., Providence (2002)
49. Millett, K.C., Rawdon, E.J.: Energy ropelength, and other physical aspects of equilateral knots. J. Comput. Phys. **186**(2), 426–456 (2003)
50. Millett, K.C., Dobay, A., Stasiak, A.: Linear random knots and their scaling behavior. Macromolecules **38**(2), 601–606 (2005)
51. Millett, K.C., Piatek, M., Rawdon, E.J.: Polygonal knot space near ropelength-minimized knots. J. Knot Theory Ramif. **17**(5), 601–631 (2008)
52. Millett, K.C., Rawdon, E.J., Tran, V.T., Stasiak, A.: Symmetry-breaking in cumulative measures of shapes of polymer models. J. Chem. Phys. **133**(15), 154113 (2010). Also cross-listed in the Virtual Journal of Biological Physics Research in the November 1, 2010 issue (volume 20, issue 9)
53. Moore, N.T., Grosberg, A.Y.: Limits of analogy between self-avoidance and topology-driven swelling of polymer loops. Phys. Rev. E, Stat. Nonlinear Soft Matter Phys. **72**(6), 061803-1-10 (2005)
54. Moore, N.T., Grosberg, A.Y.: The abundance of unknots in various models of polymer loops. J. Phys. A, Math. Gen. **39**, 9081–9092 (2006)
55. Moore, N.T., Lua, R.C., Grosberg, A.Y.: Topologically driven swelling of a polymer loop. Proc. Natl. Acad. Sci. USA **101**(37), 13431–13435 (2004)
56. Moore, N.T., Lua, R.C., Grosberg, A.Y.: Under-knotted and over-knotted polymers: 1. Unrestricted loops. In: Calvo, J.A., Millett, K.C., Rawdon, E.J., Stasiak, A. (eds.) Physical and Numerical Models in Knot Theory. Ser. Knots Everything, vol. 36, pp. 363–384. World Sci. Publishing, Singapore (2005)
57. Moore, N.T., Lua, R.C., Grosberg, A.Y.: Under-knotted and over-knotted polymers: 2. Compact self-avoiding loops. In: Calvo, J.A., Millett, K.C., Rawdon, E.J., Stasiak, A. (eds.) Physical and Numerical Models in Knot Theory. Ser. Knots Everything, vol. 36, pp. 385–398. World Sci. Publishing, Singapore (2005)
58. Orlandini, E., Whittington, S.: Statistical topology of closed curves: Some applications in polymer physics. Rev. Mod. Phys. **79**, 611–642 (2007)
59. Orlandini, E., Tesi, M.C., Janse van Rensburg, E.J., Whittington, S.G.: Asymptotics of knotted lattice polygons. J. Phys. A **31**(28), 5953–5967 (1998)

60. Plunkett, P., Piatek, M., Dobay, A., Kern, J.C., Millett, K.C., Stasiak, A., Rawdon, E.J.: Total curvature and total torsion of knotted polymers. Macromolecules **40**(10), 3860–3867 (2007)
61. Rawdon, E.J., Scharein, R.G.: Upper bounds for equilateral stick numbers. In: Physical Knots: Knotting, Linking, and Folding Geometric Objects in $\mathbb{R}^3$, Las Vegas, NV, 2001. Contemp. Math., vol. 304, pp. 55–75. Amer. Math. Soc., Providence (2002)
62. Rawdon, E.J., Dobay, A., Kern, J.C., Millett, K.C., Piatek, M., Plunkett, P., Stasiak, A.: Scaling behavior and equilibrium lengths of knotted polymers. Macromolecules **41**(12), 4444–4451 (2008)
63. Rawdon, E.J., Kern, J.C., Piatek, M., Plunkett, P., Millett, K.C., Stasiak, A.: The effect of knotting on the shape of polymers. Macromolecules **41**, 8281–8287 (2008)
64. Rivetti, C., Walker, C., Bustamante, C.: Polymer chain statistics and conformational analysis of DNA molecules with bends or sections of different flexibility. J. Mol. Biol. **280**, 058102 (1988)
65. Salzburg, Z.W., Jacobson, J.D., Fickett, W., Wood, W.W.: Application of the Monte Carlo method to the lattice-gas model. J. Chem. Phys. **30**, 65 (1959)
66. Shimamura, M.K., Deguchi, T.: Anomalous finite-size effects for the mean-squared gyration radius of Gaussian random knots. J. Phys. A **35**(18), 241–246 (2002)
67. Shimamura, M.K., Deguchi, T.: Finite-size and asymptotic behaviors of the gyration radius of knotted cylindrical self-avoiding polygons. Phys. Rev. E, Stat. Nonlinear Soft Matter Phys. **65**(5), 051802 (2002)
68. Tesi, M.C., Janse Van Rensburg, E.J., Orlandini, E., Whittington, S.G.: Monte Carlo study of the interacting self-avoiding walk model in three dimensions. J. Stat. Phys. **82**, 155–181 (1996)
69. Toussaint, G.: The Erdös-Nagy theorem and its ramifications. Comput. Geom. **31**, 219–236 (2005)
70. Valle, E., Favre, M., De Los Rios, A., Rosa, A., Dietler, G.: Scaling exponents and probability distributions of DNA end-to-end distance. Phys. Rev. Lett. **95**, 158105 (2006)
71. Vanderzande, C.: Lattice Models of Polymers. Cambridge University Press, Cambridge (1998)
72. Vologodskii, A.V., Anshelevich, V.V., Lukashin, A.V., Frank-Kamenetskii, M.D.: Statistical mechanics of supercoils and the torsional stiffness of the DNA double helix. Nature **280**, 294–298 (1979)
73. Wolfram Research, Inc.: Date and time functions. In: Wolfram Mathematica 7 Documentation. http://reference.wolfram.com/mathematica/tutorial/DateAndTimeFunctions.html. Cited Jan. 2010