

Torsional random-walk statistics on lattices using convolution on crystallographic motion groups

Aris Skliros, Gregory S. Chirikjian*

Department of Mechanical Engineering, Johns Hopkins University, 223 Latrobe Hall, 3400 North Charles Street, Baltimore, MD 21218, USA

Received 17 November 2006; received in revised form 28 January 2007; accepted 29 January 2007

Available online 2 February 2007

Abstract

This paper presents a new algorithm for generating the conformational statistics of lattice polymer models. The inputs to the algorithm are the distributions of poses (positions and orientations) of reference frames attached to sequentially proximal bonds in the chain as it undergoes all possible torsional motions in the lattice. If z denotes the number of discrete torsional motions allowable around each of the n bonds, our method generates the probability distribution in end-to-end pose corresponding to all of the z^n independent lattice conformations in $O(n^{D+1})$ arithmetic operations for lattices in D -dimensional space. This is achieved by dividing the chain into short segments and performing multiple generalized convolutions of the pose distribution functions for each segment. The convolution is performed with respect to the crystallographic space group for the lattice on which the chain is defined. The formulation is modified to include the effects of obstacles (excluded volumes) and to calculate the frequency of the occurrence of each conformation when the effects of pairwise conformational energy are included. In the latter case (which is for three dimensional lattices only) the computational cost is $O(z^4 n^4)$. This polynomial complexity is a vast improvement over the $O(z^n)$ exponential complexity associated with the brute-force enumeration of all conformations. The distribution of end-to-end distances and average radius of gyration are calculated easily once the pose distribution for the full chain is found. The method is demonstrated with square, hexagonal, cubic and tetrahedral lattices.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Lattice random-walk statistics; Motion group convolution; Pairwise torsion angle effects

1. Introduction

Exact enumeration of *all conformations* of lattice models of long polymers is a computational problem of exponential complexity. However, the generation of *conformational statistics* for this exponential number of phantom chain conformations is in fact tractable and we provide exact algorithms to compute these statistics. We present a new method for computing conformational statistics of lattice random-walk models of polymers by performing convolutions of functions on crystallographic space groups. While this method does not account

for sequentially distant interactions in the chain, it does account for local interactions within the chain, as well as all interactions of the chain with obstacles and boundaries of arbitrary shape.

In previous works, probability distributions (pdfs) of classical random walks on lattices have been computed by performing translational convolutions of “one-step” pdfs that describe allowable translations in a lattice (see e.g., [40] and references therein). However, this is *not* the problem that we are addressing. In lattice models of phantom polymer chains, a greater degree of reality can be imposed by disallowing immediate reversals and by incorporating the effects of sequentially adjacent interactions. Incorporating both these effects into overall conformational statistics requires knowledge of the local relative orientations of bonds in the chain. Therefore, translational convolutions in the lattice are not sufficient. In contrast,

* Corresponding author. Tel.: +1 410 516 7127; fax: +1 410 516 7254.

E-mail addresses: askliro1@jhu.edu (A. Skliros), gregc@jhu.edu (G.S. Chirikjian).

convolution on space groups (which include the effects of local orientational changes) becomes a powerful tool.

In this section we provide a brief review of related literature, the torsional random-walk model that will be used, and the limitations of brute-force enumeration. This provides the necessary background that will be required before pursuing the space-group convolution method for computing lattice conformational statistics. Section 2 will describe the space-group convolution method in detail for phantom torsional random walks. Section 3 discusses some important details for the lattices of most common interest. Section 4 modifies the basic approach to account for obstacles and pairwise torsional energetic effects. Section 5 presents numerical results. This is followed by a discussion of computational performance and our conclusions.

1.1. Literature review

Random and self-avoiding walks on lattices arise in a number of fields. One notable example is polymer theory. The study of the statistical behavior of long polymer chains on lattices began approximately 50 years ago, shortly after the introduction of the Monte Carlo sampling method [16]. Polymer science at that time had already existed for decades [7]. Several models were developed through the decades of the 1950s and 60s, which are summarized in Flory's classic book [5]. The study of the statistics of polymers [8–10] became a matter of interest because of the wide range of properties that they have. Polymers have many interesting and useful mechanical properties, which are due to their molecular structure [6]. Some of these properties, such as rubber elasticity [6] and the toughness and ductility of semi crystalline polymers, depend on the ensemble behavior of polymer chains that exist in a wide variety of conformations [6]. It is therefore a matter of importance to study the structure and statistics of ensembles of long polymer molecules. The size of a polymer chain is one of the fundamental quantities in the study of its structure. The mean-square end-to-end distance of a polymer chain is one description of the size of a polymer chain [6].

A polymer molecule is composed of monomers which typically consist of a central carbon atom and atoms of other elements such as N, H, etc. These monomers are connected with each other by covalent chemical bonds, thus forming the polymer chain. Let \mathbf{r}_{i-1} denotes the position of the central carbon of the i th monomer and $i = 1, \dots, n + 1$ enumerate the $n + 1$ monomers. Then the bond vector, $\mathbf{b}_i = \mathbf{r}_i - \mathbf{r}_{i-1}$, connects two central carbon atoms of sequential monomers. Summing the n bond vectors of $N + 1$ identical monomers in the chain we find the end-to-end distance vector \mathbf{r} [6]:

$$\mathbf{r} = \sum_{i=1}^{n+1} \mathbf{r}_i - \mathbf{r}_{i-1}. \quad (1)$$

The corresponding end-to-end distance is $r = |\mathbf{r}|$. The mean-square end-to-end distance $\langle r^2 \rangle$ is very important in order to understand the structure of the polymer [6]. Likewise, other important physical quantities are related to the distribution

of values of r , denoted here as $p(r)$. Many simulation methods have been developed in order to obtain information about how $p(r)$ evolves for polymer chains. These methods have been applied to many models of polymer chains. Our study focuses on the Random Walk (RW) on lattices.

Let Δ_r^2 denotes the relative mean-square fluctuation of r^2 defined by $\Delta_r^2 = (\langle (r^2)^2 \rangle - \langle r^2 \rangle^2) / \langle r^2 \rangle^2$ [6]. A serious problem in computer simulation of polymers is the lack of self-averaging:

$$\Delta_r^2 \not\rightarrow 0 \quad \text{as } n \rightarrow \infty, \quad (2)$$

but it is desirable to have $\Delta_r^2 \rightarrow 0$ as $n \rightarrow \infty$ [6]. Eq. (2) has led many researchers to use simplified models such as lattice random walks [18,19], and therefore estimate more closely the real values of $\langle r^2 \rangle$. These values can be approximated well only for simple lattice models for a high number of independent conformations [20,15]. These methods started with the pioneering work of Wall and collaborators [13,31] on single chains and subsequently were extended to more complex systems [12,14]. Furthermore analytical theories using lattice models [16,21,22,6] are based on a lattice descriptions [23,6], such as excluded volumes or Self-Avoiding Walks (SAWs) [11]. The computational cost for brute-force enumeration is high, hence not practical [24–28,11]. In any case, lattice models are ideal for sampling methods and algorithms such as the slithering snake algorithm [29,30], the pivot algorithm [32,33], the conformation bias algorithm [33,6] and the chain breaking algorithm [34,35]. The main purpose of these algorithms is to approximate the true polymer conformation. One characteristic of polymer chains is that repulsive forces do not allow them to self-intersect or two different monomers to occupy the same position in space [6]. So an improved model of polymer simulation is the Non-Reversal-Random-Walk (NRRW) [17], in which immediate reversals are not allowed but sequentially nonadjacent self-intersections are allowed.

A subtle distinction exists between NRRWs and more kinematically realistic models of polymers in which only rotational (torsional) moves are allowed, as is shown in Fig. 1. For example, in a cubic lattice a random walk can move in six directions. The NRRW would allow five directions (all but the direction pointing backwards along the direction of the current move). The torsional model would allow only four directions (those which are orthogonal to the direction of the current move). The distinction between NRRW and torsional models is really only important for square and cubic lattices, since there is no way for consecutive bonds to be parallel in the hexagonal and tetrahedral lattices. If the number of rotational moves available around each bond vector is z , then the total number of conformations that can be generated by a torsional random-walk model is z^n [36]. Another issue that arises in polymer simulations is how the presence of excluded volume changes the distribution of the end-to-end distance. Monte Carlo simulations have been used to deal with these problems [41,42].

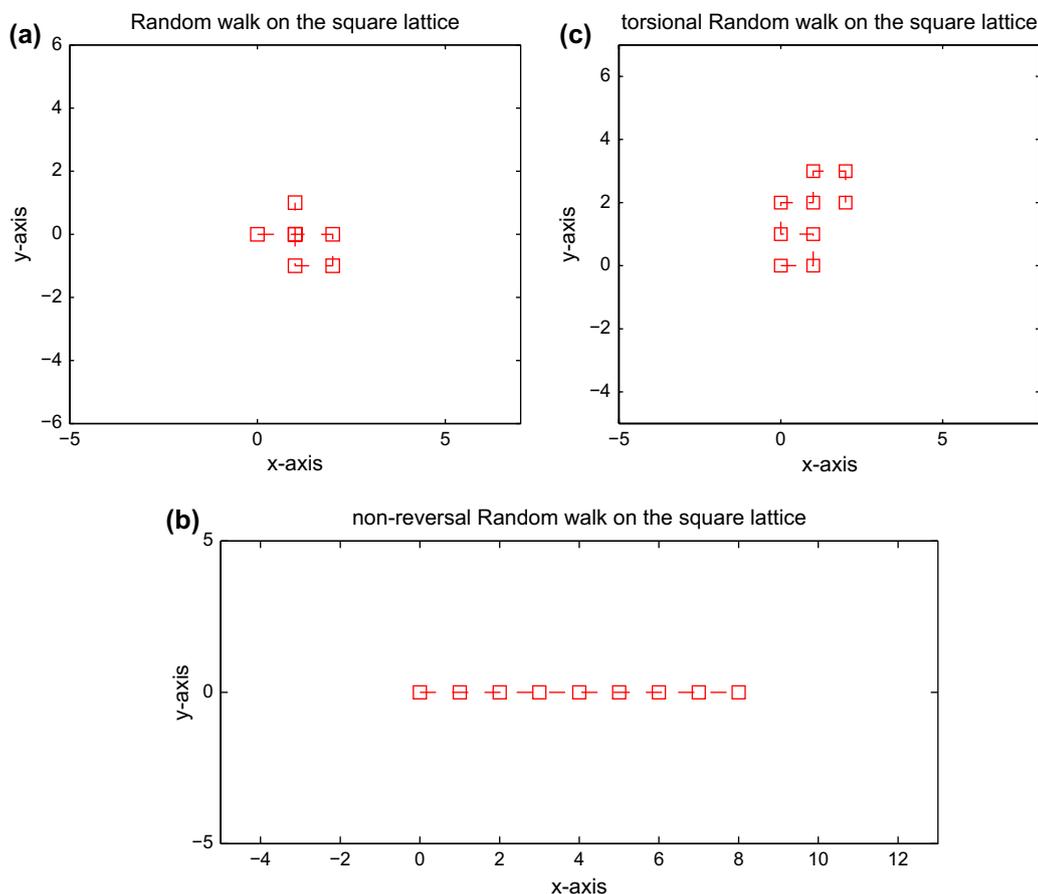


Fig. 1. A schematic description of lattice walks and attached reference frames: (a) random walk; (b) non-reversal; (c) torsional.

1.2. Overview of the torsional random-walk model

Let us denote the value of the bond angles (which are always constant) as $\theta_i = \theta$ for $i = 1, \dots, n$. For the cubic and square lattice the bond angle θ is 90° , for the hexagonal lattice it is 120° , and for the tetrahedral lattice it is 109.8° . The torsion (dihedral) angles, $\{\phi_i\}$ in the cubic and tetrahedral lattices have discrete values $0^\circ, 90^\circ, 180^\circ, 270^\circ$ and $0^\circ, 120^\circ, 240^\circ$, respectively. Each of these 4 and 3 rotations represent a subgroup of the point groups for the cube and of the tetrahedron, respectively. As one progresses along the chain, the change in orientation imposed by bond angles causes a ‘mixing’ of these subgroups until all of the elements of the point group (group of rotational symmetry operations) are realized. For the planar lattices we do not have torsion angles. To each site of the lattice we assign, apart from the Cartesian coordinates, a number which corresponds to the orientation of the end of the random walk with respect to the origin.

The way that polymer chains are modeled in the lattice is that each lattice site is considered as the carbon atom of a monomer, whereas the lattice segment (lattice edge) corresponds to the polymer bond. The number of random walks generated by an n -segment random walk increases exponentially w.r.t n . It is therefore impossible to enumerate directly all the conformations of a long polymer. We therefore need some other methods. Popular methods that have been used

in the past and have been mentioned before are based on random sampling. Although more sophisticated algorithms than simple sampling have been proposed [4,38,39] these methods do not capture the tails of certain kinds of pdfs in polymer science [1].

Therefore a new method which is cost effective and does not have the drawbacks of the previous methods is needed. Such a method is presented in this paper: the method of crystallographic space-group convolution. This method generates the exact *probability distributions* for all the possible z^n conformations of n -segment lattice chains corresponding to torsional random walks. This is done without having to pay the exponential cost of exhaustive enumeration. We use the torsional random-walk model, at $O(n^3)$ computational cost for two dimensional lattices (hexagonal, square) or $O(n^4)$ for three dimensional lattices. We present an example here to illustrate the great potential of this method. By using the crystallographic space-group convolution we can find, for the case of the square lattice, the distribution describing where *all* of the $2^{50} = 1.12 \times 10^{15}$ torsional random walks of length 50 terminate. This is computed in 9.74 s on a personal computer. Fig. 2 shows the distribution, where darker dots mean more walks terminating on that lattice site. In addition to that we can use this method to find where all the possible walks of 50 segments terminate if we have an obstacle. For our case the obstacle is a circle with center $(-10, 14)$ and radius 8. It

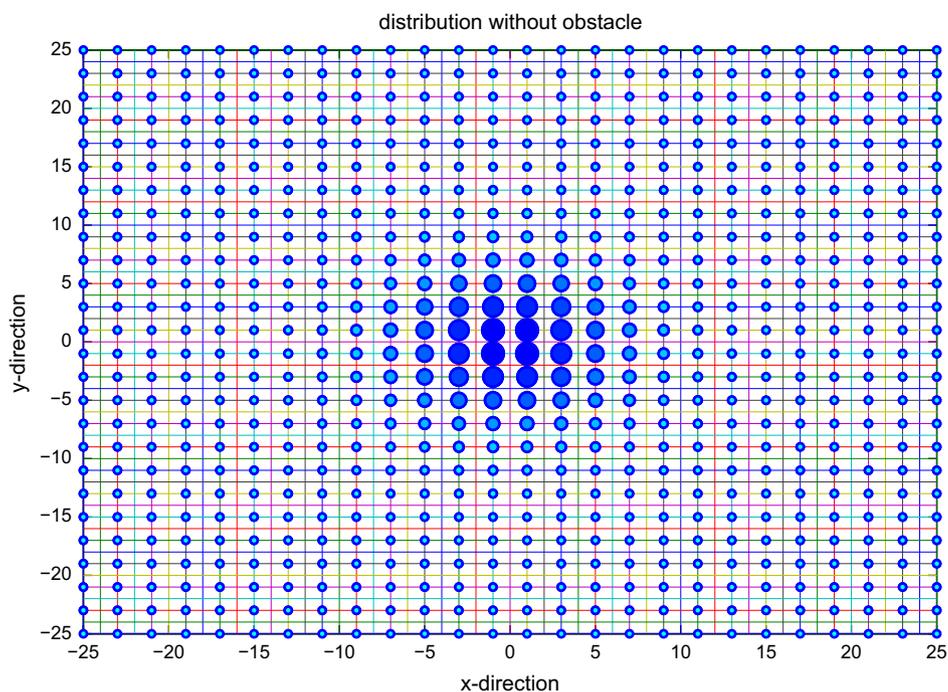


Fig. 2. Distribution of the end of *all* the 50-segments random walks.

is worth mentioning that due to the existence of this obstacle the number of walks of 50 segments is much less. For this example, the number of walks when there is an obstacle is 1.0827×10^{15} . We normalize again over 2^{50} . The results are shown in Figs. 2 and 3 to compare the distributions in these two cases.

It is important to note that we generate probability distributions in pose (position and orientation) for all torsional random walks of length 1 through n using our method. And whereas statistics for rotationally isotropic walks can be computed using *translational* convolution in lattices, this is *not* what we are doing. In torsional random walks, small local changes cause

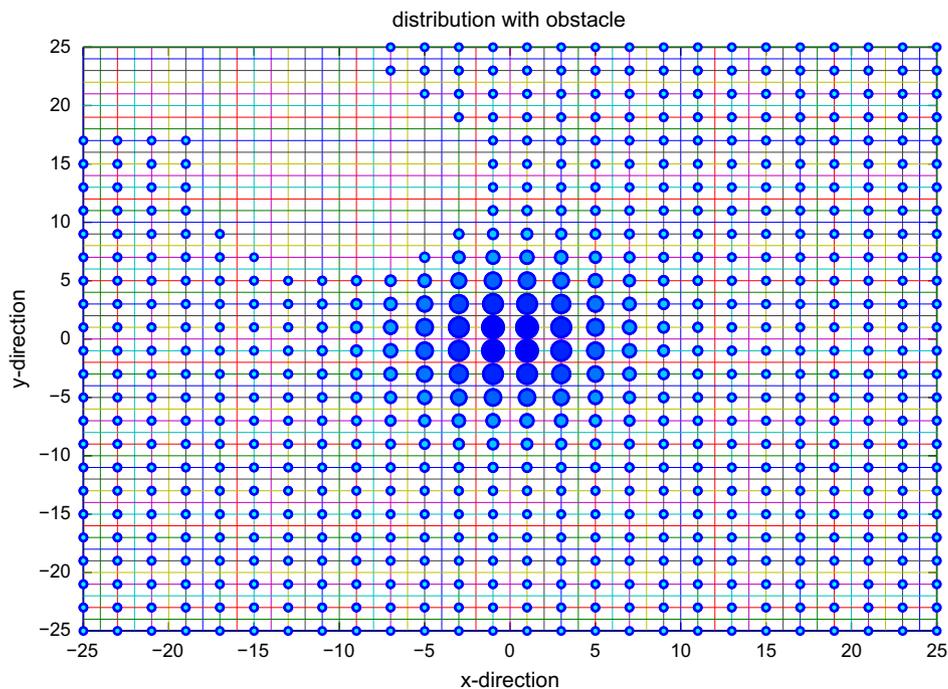


Fig. 3. Distribution of the end of *all* the 50-segments random walks with obstacle.

large positional and orientational changes at distant points in the chain. Hence, full *space-group* convolution is a more natural tool.

Attached to each individual random walk is a set of reference frames. The origin of each of these frames is located on a lattice site that has a certain position w.r.t. x,y,z Cartesian coordinates. Each frame also has a certain orientation w.r.t. an inertial frame fixed at the origin. That orientation corresponds to one of the rotational symmetries of the polygon or the polyhedron (i.e. the unit cell) from which the lattice is constructed. All combinations of allowable translations and rotations form a crystallographic space group.

Let G be the crystallographic space group of a lattice with operation \circ , and let g_i denote an arbitrary element of G . Each g_i can be represented as a homogeneous transformation matrix consisting of a rotation R_i and translation \mathbf{t}_i

$$g_i = g(R_i, \mathbf{t}_i) = \begin{bmatrix} R_i & \mathbf{t}_i \\ 0^T & 1 \end{bmatrix}. \quad (3)$$

Two rigid-body motions are composed as:

$$g_1 \circ g_2 = g(R_1, \mathbf{t}_1) \circ g(R_2, \mathbf{t}_2) = g(R_1 R_2, R_1 \mathbf{t}_2 + \mathbf{t}_1) \quad (4)$$

which is viewed clearly if we perform the matrix multiplication of g_1 and g_2 , respectively. The inverse of g_i is defined as:

$$g_i^{-1} = \begin{bmatrix} R_i^T & -R_i^T \mathbf{t}_i \\ 0^T & 1 \end{bmatrix}. \quad (5)$$

The number of elements in space groups is infinite simply because of the fact that the lattice dimensions can be extended to infinity. However, it is possible to “periodize” the lattice to make it finite, and this will not cause problems as long as the period of the lattice is sufficiently large relative to the length of the polymer. On the other hand, the number of elements R_i in the group of rotational symmetries is always finite. Since each g_i describes “position and orientation”, the word “pose” will mean any element g_i .

1.3. The limitations of brute-force enumeration

In the brute-force case if one wants to move from the n th frame attached to an n -segment random walk which is described by the matrix g_n to the $n+1$ -st frame of an $n+1$ -segment random walk then one could proceed as follows. First find all the possible poses that the 1-segment random walk on the lattice can reach using allowable torsional moves. The number of these is taken to be 2 for the square and hexagonal lattices, 3 for the tetrahedral and 4 for the hexagonal and cubic when using the torsional model. As an example, consider the tetrahedral lattice. The transformations defining local moves are given by the matrices, $g^{(1)}$, $g^{(2)}$, $g^{(3)}$, that are described with matrices of the same form as in Eq. (3), but we use superscripts in place of subscripts to denote that these are local transformations. In order to find all of the possible 2-segment NRRWs on the tetrahedral lattice we multiply each of these transformations with each other, so we have nine

matrix multiplications. For the 3-segment random walk case we have 27 multiplications and for the N -segment case we have 3^n multiplications. So for finding all the possible poses for the $n+1$ -st lattice site of the $n+1$ -segment random walk, brute-force enumeration would take the matrix products $g_n \circ g^{(i)}$, for $i = 1, 2, 3$ for each of the 3^n values of the poses g_n . The orientation of this new pose will have the matrix representation $R_n R^{(i)}$ and the position will be $R_n \mathbf{t}^{(i)} + \mathbf{t}_n$. These are the three possible $n+1$ -segment random walks on the lattice generated by adding to each of the preceding n -segment random walks. The same procedure is followed for all the other lattices. Clearly this is an exponential problem.

While the above approach is conceptually simple, the problem is that it is computationally not practical. In order to circumvent the exponential complexity of brute-force enumeration, this paper consists of the development of a convolution for the square, the hexagonal, the cubic and the tetrahedral lattices. For the square lattice we generate statistical quantities exactly without enumerating exhaustively all the NRRWs for up to $n = 1023$ (hence 8.9885×10^{307}) conformations and we present the distributions of r . Then we present the distribution of r values for one obstacle (in the polymer case obstacles can be monomers of other chains or they can represent impenetrable boundaries [37]). This is done for $n = 50, 100, 150, 200$ with and without accounting for the obstacle. For the hexagonal lattice we do the calculations for $n = 50, 100, 150, 200$ with and without accounting for the obstacle. For the cubic lattice we have two cases when we do not account for the obstacles. When the effects of pairwise conformational energy terms are disregarded and when they are taken into account. For the former case we find exact statistical quantities for all possible 4^{50} , 4^{100} possible NRRWs in 220 s on a PC and we present the distribution of r . Then we modify the convolution for three different obstacles and we present again the distribution of r . In the case when the effects of conformational energy are taken into account the computational cost is multiplied by $16^2 = 256$. This will be clear by the analysis of the cubic lattice, where we present the same distribution as before and we observe the differences. When accounting for the obstacles we make the calculations for $n = 15$ without accounting for torsion angles. For the tetrahedral lattice for the case that we do not have obstacles we make the calculations for $n = 50, 100$ when we are not taking into account the torsion angles and for $n = 30$ when we are. We calculate the distribution of r when we have obstacle for $n = 50$, without accounting the torsion angles. When taking into account the affects of conformational energy the cost is multiplied by $3^4 = 81$.

The new feature of this method is that it can generate exact statistics (to within machine precision) corresponding to *all* of the exponential number of conformations in a polynomial amount of time without having to resort to sampling only a small subset of conformations. This becomes more and more convenient when the number of segments increases. While previous methods relied on sampling, the new method finds statistics for all the possible conformations in a very short time.

2. Overview of the space-group convolution method

Firstly we begin by explaining the geometric intuition of the method of generalized convolution of functions on space groups of a discrete lattice assuming that we have three frames of reference O_1 , O_2 and O_3 with origins located at lattice points, and orientations that are elements of corresponding point groups. We view O_1 as being fixed, O_2 as moving with respect to O_1 , and O_3 as moving with respect to O_2 , as in Fig. 4 [1–3,39].

Assume that g_1 is the homogeneous transformation (i.e., a matrix of the form in Eq. (3)) that describes the pose (position and orientation) of O_2 with respect to O_1 , g_2 is the one that describes the pose of O_3 with respect to O_2 . Then the pose (position and orientation) of O_3 with respect to O_1 is described by $g_3 = g_1 \circ g_2$. Then $g_2 = g_1^{-1} \circ g_3$. At each lattice site is attached a frame of reference. On the n -segment random walk the original lattice site is the one to which the inertial frame (identity motion) is attached. Working on lattices so, ensures us that we move g_1 and g_2 over a finite number of different poses. Since g_1 and g_2 move through a finite number of different poses, we can define f_{n_1} as a function which assigns as a value the frequency of occurrence of the distal end of a walk of length n_1 reaching g_1 , while the proximal end is rooted at the inertial frame. Similarly f_{n_2} can be defined as the function which takes as a value the frequency of g_2 . In general $f_{n_i}(g)$ denotes a space-group function, where the subscript n_i stands for the number of segments of that particular random walk. It takes as an input the position and orientation g and gives as an output the number of times that position and orientation is reached by the distal end of an n_i -segment random walk on that specific lattice. Our main target is to derive the distribution of g_3 frames based on the knowledge of g_1 and g_2 . The methodology followed is the natural outcome of the reasoning described above and proceeds as follows. Step 1: evaluation of $f_{n_1} = f_{n_1}(g_1)$. Step 2: evaluation of $f_{n_2} = f_{n_2}(g_2) = f_{n_2}(g_1^{-1} \circ g_3)$. Step 3: weighted multiplication of $f_{n_2}(g_1^{-1} \circ g_3)$ by the number of frames at g_1 , which is $f_{n_1}(g_1)$. Step 4: sum over all of these contributions

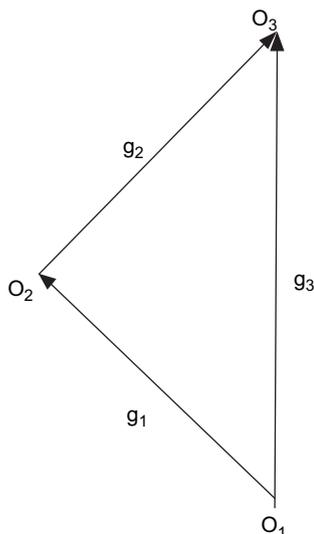


Fig. 4. Frames of reference.

$$f_{n_1+n_2}(g_3) = (f_{n_1} * f_{n_2})(g_3) = \sum_{g_1 \in G} f_{n_1}(g_1) f_{n_2}(g_1^{-1} \circ g_3). \quad (6)$$

Here G denotes the full crystallographic space group, but in practice since the functions f_{n_i} diminish to zero outside of a small range, the sum is taken over a finite number of group elements. Since each sub-chain has z^{n_1} and z^{n_2} states, respectively, the set of all states of the combined walks of length $n_1 + n_2$ is $z^{n_1} \cdot z^{n_2} = z^{n_1+n_2}$. The function resulting from the convolution of $f_{n_1}(\cdot)$ and $f_{n_2}(\cdot)$, respectively is the distribution function for the whole ensemble $f_{n_1+n_2}(g_3) = (f_{n_1} * f_{n_2})(g_3)$. Moreover, this way of thinking can be applied to walks that emerge from composing more than two walks stacked on top of one another. For example if for walks that are stacked the distribution of the lower two is given by $f_{n_1} * f_{n_2}$ and the distribution of the upper by $f_{n_3} * f_{n_4}$, then the distribution of the concatenation of segments is given by $f_{n_1+n_2+n_3+n_4} = (f_{n_1} * f_{n_2}) * (f_{n_3} * f_{n_4}) = f_{n_1} * f_{n_2} * f_{n_3} * f_{n_4}$ because the operation of convolution is associative. For $n_1 + n_2 + \dots + n_K$ segments it holds that $f_{n_1+n_2+\dots+n_K}(g_3) = (f_{n_1} * f_{n_2} * \dots * f_{n_K})(g_3)$. This last equation shows the procedure that we will follow in our codes to find the probability distribution of an $n_1 + n_2 + \dots + n_K$ -segment random walk, where f_{n_i} for $i = 1, \dots, K$ is the function of n_i -segment random walk, where n_i is an arbitrary number of segments [1–3,39].

If one compares the computational complexity of this method with the brute-force enumeration of ensemble states one concludes that this method is computationally much more efficient. To justify this conclusion, we begin by assuming an arbitrary chain consisting of K sub-chains (each one consisting of an arbitrary n_i , $i = 1, 2, \dots, K$ number of segments). The computational cost of enumerating conformations of each sub-chain is z^{n_1}, \dots, z^{n_K} . Brute-force enumeration of all the conformations of the total chain is $O(z^{n_1+n_2+\dots+n_K})$. In contrast in our new approach, functions f_{n_i} are used for the description of the distribution for each sub-chain f_{n_1}, \dots, f_{n_K} at a cost of $O(\sum_{k=1}^K z^{n_k})$. To find the frame distribution of the whole ensemble we perform $K - 1$ convolutions [1–3,39].

The cost of these convolutions depends upon the dimension of the lattice, i.e., $D = 2, 3$. The number of lattice points reachable by the distal end of an n -segment random walk is $O(n^2)$ and $O(n^3)$ for planar and spatial lattices, respectively. By considering the convolution of two adjacent functions we have that the numerical computation of the convolution sum evaluated at a single point in the support of $(f_{n_i} * f_{n_{i+1}})$ becomes a sum over all lattice motion group elements in the support of f_{n_i} . This calculation must be performed for all lattice motion group elements in the support of $f_{n_i} * f_{n_{i+1}}$, so all computations required are $O(\text{convolution}) = O(Q_{n_i} \cdot Q_{n_{i+1}})$. Here Q_{n_i} and $Q_{n_{i+1}}$ are the number of lattice motion group elements in the support of f_{n_i} and $f_{n_{i+1}}$, respectively. Since $Q_{n_i} = O(n_i^D)$ and D is a constant, the computational cost of $K - 1$ convolutions is a polynomial in K . The order of this polynomial depends on D [1–3,39].

In our implementation the chain is broken into n individual bonds, i.e., $K = n$ and $n_i = 1$. Thus the frame distribution $f_n(g_3)$ is calculated from n identical “one-step walk” functions as an n -fold convolution $f_n = f_1^{(n)} = f_1 * f_1 * \dots * f_1$. The following

series of convolutions are performed until we reach the desired result $f_1^{(2)} = f_1 * f_1, f_1^{(3)} = f_1^{(2)} * f_1, \dots, f_1^{(n)} = f_1^{(n-1)} * f_1$. Q_i which is the number of points in the support of f_1 is always a constant independent of n so the total computational cost becomes $c_1 \cdot 1^D + c_2 \cdot 2^D + \dots + c_{n-1} \cdot (n-1)^D = O(n^{D+1})$ [1–3,39].

The above approach is not the only one that can be taken. For example, it is possible to compute a cascade of the form $f_2 = f_1^{(2)}$ and then $f_4 = f_2 * f_2, f_8 = f_4 * f_4$, etc. using $\log_2 n$ convolutions. Each of these convolutions can be computed using the FFT for periodized lattices in $O(P^D \log P)$ arithmetic operations where the number of sample points in the support of the resulting function is $P = O(n_i)$ in each translational direction and n_i is the length of the resulting walk. However, in practice iteratively performing one-step convolutions is faster and allows us to handle obstacles. This is not true for the cascaded approach described above [1–3,39].

Having two frame functions the definition of the convolution in Eq. (6) can be written for the spatial case as:

$$f_{n_1+n_2}(i', j', k', m') = \sum_{i,j,k,m} f_{n_1}(i, j, k, m) \cdot f_{n_2}(i'', j'', k'', m'') \quad (7)$$

where i, j, k, i', j', k' stand for the position, whereas m, m' stand for the orientation. For the planar case we have i, j, i', j' for the position and k, k' for the orientation. By combining Eqs. (5)–(7) we see that:

$$\begin{bmatrix} i'' \\ j'' \\ k'' \end{bmatrix} = R_i^{-1}(\mathbf{t}_j - \mathbf{t}_i). \quad (8)$$

3. Example lattices

In this section we examine some of the details required to implement the space-group convolutions for torsional random walks in four different kinds of lattices: square, hexagonal, cubic and tetrahedral.

3.1. Square lattices

The 1-segment torsional random walk in the square lattice has two possible poses at its distal end relative to its proximal end (see Fig. 5). For the 2-segment torsional RW we will have $2^2 = 4$ possible poses and for the k segment torsional random walk we have 2^k possible poses. The one-step motions and elements of the space group for the unit square lattice are, respectively, of the form:

$$g^{(m)} = \begin{bmatrix} 0 & -m & -m \\ m & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } g = \begin{bmatrix} \cos(\frac{k\pi}{2}) & -\sin(\frac{k\pi}{2}) & i \\ \sin(\frac{k\pi}{2}) & \cos(\frac{k\pi}{2}) & j \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

where $m = \pm 1, k = 0, 1, 2, 3$ and i, j are independent integers. Applying convolution of functions of motion on the square lattice we just have to follow the notation of Eqs. (6) and (7), by plugging in the elements of the square lattice group.

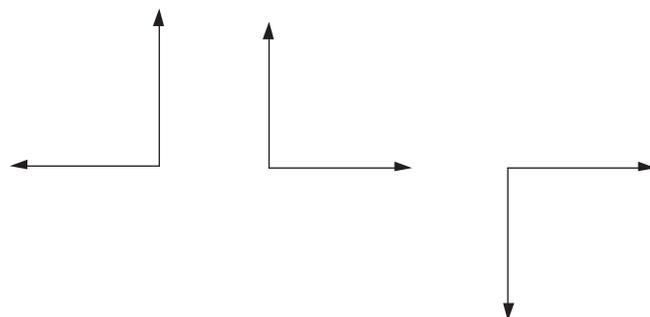


Fig. 5. Evolution of the two segments on the square lattice.

3.2. Hexagonal lattices

We do the same thing for the hexagonal lattice (see Fig. 6): we consider the length of each edge of the hexagons on the lattice to be unity. First we have to find the rotation matrices which represent the symmetries of the hexagon. These are given by $R_z(k\pi/3)$ for $k = 0, 1, 2, 3, 4, 5$ where $R_z(\alpha)$ denotes counterclockwise rotation around the z -axis by angle α . The matrices $R_z(k\pi/3)$ describe the planar rotational symmetries of the hexagon. For the planar hexagonal lattice the one-step motions and elements of the space group are, respectively, of the form:

$$g^{(m)} = \begin{bmatrix} \cos(\frac{m\pi}{3}) & -\sin(\frac{m\pi}{3}) & -m\frac{\sqrt{3}}{2} \\ \sin(\frac{m\pi}{3}) & \cos(\frac{m\pi}{3}) & \frac{1}{2} \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

and $g = \begin{bmatrix} \cos(\frac{k\pi}{3}) & -\sin(\frac{k\pi}{3}) & \frac{\sqrt{3}}{2}i \\ \sin(\frac{k\pi}{3}) & \cos(\frac{k\pi}{3}) & \frac{1}{2}j \\ 0 & 0 & 1 \end{bmatrix}$

for $m = \pm 1, k \in [0,5], i, j$ are integers. For computing i we have to know the number of segments this walk refers to. If n is the number of segments then $i = -n:n$. For j , things are slightly more complicated. For n even and $n > 3$ the minimum

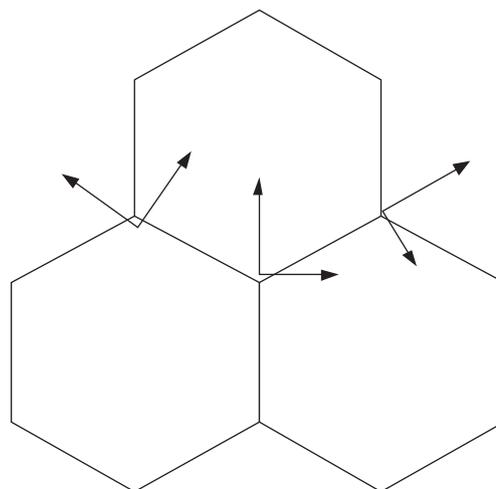


Fig. 6. Evolution of the two segments on the hexagonal lattice.

and the maximum values of j are $(-3n + 6)/2$ and $3n/2$, respectively. For n odd and $n > 3$ the minimum and the maximum values of j are $(-3n + 4)/2$ and $(3n - 1)/2$, respectively. The rest of the values of j are the values reached by $m < n$ -segment random walks. For $n = 1, 2, 3$ the minimum and maximum values for j are 1, 0, -2 and 1, 3, 4, respectively. To compute the convolution on the hexagonal lattice we just have to follow the notation of Eqs. (6) and (7), by substituting the elements of the hexagonal lattice group.

3.3. Cubic lattices

For the cubic lattice case (see Fig. 7), the rotation matrix describing the orientation of a frame attached to bond vector $i + 1$ relative to the frame attached to bond i is written as:

$$R_i = \begin{bmatrix} -\cos \theta_i \sin \phi_i & \sin \phi_i & \sin \theta_i \cos \phi_i \\ -\cos \theta_i \cos \phi_i & -\cos \phi_i & \sin \theta_i \sin \phi_i \\ \sin \theta_i & \cos \theta_i & 0 \end{bmatrix} \quad (11)$$

where $\theta_i = -\pi/2$.

The elements g_i of the cubic lattice space group are given by:

$$g_i = \begin{bmatrix} R_i & e_3 \\ 0^T & 1 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad (12)$$

or

$$g_i = \begin{bmatrix} 0 & \sin \phi_i & -\cos \phi_i & 0 \\ 0 & -\cos \phi_i & -\sin \phi_i & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

for $\phi_i = 0, \pi/2, \pi, 3\pi/2$.

The 24 rotational symmetry operations for the cube are well known. The four poses of the 1-segment random walk

for the cubic lattice are found as follows. We first translate along the z -axis by one unit, then we rotate around the new y -axis (which happens to be the same with the previous y -axis) by $\pi/2$ and then we rotate around the new x -axis either by 0, or $\pi/2$, or π , or $3\pi/2$. We find the function of n -segment random walk by using the technique of the crystallographic convolution in the cubic Lattice. Again the definition of the convolution in the cubic lattice is achieved by plugging in the elements g_i of the cubic lattice crystallographic space group into Eq. (6), where $n_1 = 1, n_2 = 1, 2, \dots, n - 1$.

3.4. Tetrahedral lattices

The tetrahedral lattice has as vertices centers of tetrahedra. Each lattice site in the tetrahedral lattice is adjacent to four other lattice sites. So proceeding from the n th lattice site of an n -segment random walk to the $n + 1$ -st lattice site of the $n + 1$ -segment random walk we have four possible choices which reduce to three for the sake of not allowing immediate reversals. The group elements of the tetrahedral crystallographic space group are given as in Eq. (3). The matrices R_i are the rotational symmetry operations for the tetrahedron of which these are 12. The Cartesian coordinates of the tetrahedral lattice will be integers if we inscribe the tetrahedron in a cube of edge length 2, see figure [49]. For finding the representations of the 12 symmetries of the tetrahedron we take the 3×3 identity matrix as the representation of the identity rotations. Then we rotate around the three vectors, which connect the origin with the centers of the equilateral triangles, which the vertices of the original tetrahedron form, about $2\pi/3, 4\pi/3$. By the term original tetrahedron we mean the tetrahedron which we inscribe in the cube. Thus we take the first 7 rotations of the tetrahedron. By combining the seven rotations with each other we obtain the remaining five. The origin is the center of this tetrahedron. The crystallographic motion group convolution on the tetrahedral lattice is defined as in Eq. (7) over the space-group elements of the tetrahedral lattice. We have to note that

$$\begin{bmatrix} i'' \\ j'' \\ k'' \end{bmatrix} = \pm R_i^{-1}(\mathbf{t}_j - \mathbf{t}_i) \quad (14)$$

due to the fact that the tetrahedral lattice is two definable [50]. If i, j, k, m of Eq. (7) refer to a function of odd number of segments then we have the negative sign in Eq. (14), otherwise we have the positive one. This can be explained as follows. The tetrahedral lattice sites are centers of one tetrahedron and vertices of another one. The tetrahedron for which the lattice site is the center looks down(up), whereas the tetrahedron for which the lattice site is a vertex looks up(down). This means that the tetrahedral lattice is periodic for every two segments and subsequently for every even number of segments thus for f_{n_i} having an even number of segments we have the positive sign holding in Eq. (14). For odd number of segments we have

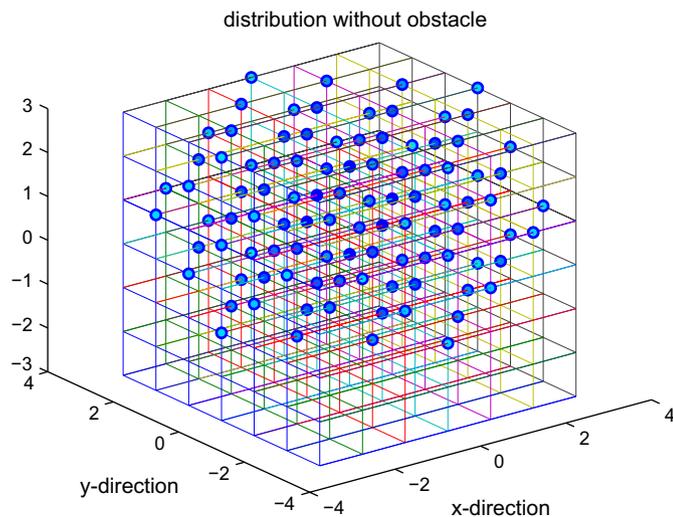


Fig. 7. Cubic lattice.

the opposite orientation every time, which is why we have negative sign.

For the case of the tetrahedral lattice the elements g_i of the spatial lattice motion group are given by:

$$g_i(\phi_i, \theta_i) = \begin{bmatrix} -\cos \phi_i & -\sin \phi_i \cos \phi_i & \sin \phi_i \sin \theta_i & 0 \\ \sin \phi_i & -\cos \phi_i \cos \theta_i & \cos \phi_i \sin \theta_i & 0 \\ 0 & \sin \theta_i & \cos \theta_i & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

where $\theta_i \approx 109.8^\circ$.

4. The effects of obstacles and pairwise conformational potential energy

In this section we modify the basic space-group convolution approach presented in Section 2 to generate probability distributions in the case when there are obstacles and pairwise energy effects between sequentially adjacent bonds.

4.1. Accounting for obstacles

One of the goals of this paper is to find the frame distribution function for the distal end of a lattice random walk for all the lattices of interest (square, hexagonal, cubic and tetrahedral) with and without obstacles. This is achieved by defining the borders of the obstacle on the lattice. When the frame distribution function of the n -segment random walk is convolved with that of a 1-segment random walk the frame distribution function of the $n + 1$ -segment random walk results. Then we examine whether the result lies in the excluded volume. If it does, we set the value to zero at that point. To make it more clear, Eq. (6) is modified as:

$$f_{n+1}(g) = \sum_h f_n(h) f_1(h^{-1} \circ g). \quad (16)$$

Note that in Eq. (16), we calculate the convolution over the support of f_n . That is because for the obstacle case the order matters. A simple example is that while $g = (i', j', k', m')$ (see Eq. (7)) may lay far beyond the obstacle, yet $h^{-1} \circ g$ may lay inside it. Thus to ascertain the validity of our calculations we set $h^{-1} \circ g$ to be the support of f_1 . When the values of i', j', k' lay in or on the obstacle then the result of the convolution given by Eq. (16) is set to zero at those values. If this is done one step at a time, the full set of obstacle-avoiding conformations will be generated exactly. Note that this is different than nullifying the value of an n -fold convolution at the end. The computational cost of this calculation is high. That is because the elements of the support of f_{n+1} are bound by a cube of volume $(2n + 3)^D$, while the elements of the support of f_n are bound by a cube of volume $(2n + 1)^D$. Thus we need to find a more efficient way of dealing with the obstacles case. The way to do this is the following: a convolution sum of the form above can be written in the following equivalent ways:

$$(f_n * f_1)(g) = \sum_{z \in G} f_n(z^{-1}) f_1(z \circ g) = \sum_{k \in G} f_n(g \circ k^{-1}) f_1(k) \quad (17)$$

where the substitutions $z = h^{-1}$ and $k = h^{-1} \circ g$ have been made, and the invariance of summation under shifts and inversions is used. In other words, we use the general facts that

$$\sum_{h \in G} f(h) = \sum_{h \in G} f(h^{-1}) = \sum_{h \in G} f(g \circ h) = \sum_{h \in G} f(h \circ g)$$

for any $g \in G$. This is true because we sum over *all* the elements of the group. In our formulation, we seek to compute $f_{n+1}(g)$ from $f_n(g)$ and $f_1(g)$. Due to the invariance of summation under the transformations discussed above, we can use the second equality in Eq. (17) to write:

$$(f_n * f_1)(g) = \sum_{k \in G} f_n(g \circ k^{-1}) f_1(k).$$

In other words, we can still use the small nature of the support of f_1 for computational advantage. This is important because when computing convolutions in the presence of obstacles, we will use this form. Thus the computational cost for the distribution functions of all torsional walks of length up to n is still $O(n^{D+1})$.

In Section 5 of this paper we find the frame distribution function for the distal end of the chain and the function $p(r)$ when we have an excluded volume and compare it to the corresponding distribution function when there is no excluded volume. For the square and hexagonal lattices the obstacle in the numerical examples will be a circle of radius 8 and center $(-10, 14)$. For the three dimensional lattices the excluded volumes would be a sphere with center $(-10, 14, 13)$ and radius 8 for the tetrahedral lattice and a sphere with center $(-5, 5, 4)$ and radius 4 for the cubic lattice. A spherical obstacle was also used in Ref. [42].

4.2. Pairwise conformational energy

If one were to consider all self-interactions within a polymer modeled as a lattice torsional walk, the probability distribution of end frames would be written as:

$$\tilde{f}_E(g) = f_E(g)/Z \quad \text{where} \quad f_E(g) = \sum_{\phi \in I(g)} \exp\left(\frac{-E(\phi)}{k_B T}\right). \quad (18)$$

Here ϕ is the vector of torsion angles, $E(\phi)$ is a conformational energy function, and $I(g)$ denotes the finite set of all torsion angles (see Fig. 8) that defines chains rooted at the global reference frame and ending at g . This is a Boltzmann-weighted distribution where k_B is the Boltzmann constant and T is the absolute temperature measured in Kelvin. The partition function is defined as:

$$Z = \sum_{\phi} \exp\left(\frac{-E(\phi)}{k_B T}\right). \quad (19)$$

For pairwise potentials one does not enumerate all conformational states by brute force to compute Z . Instead, Flory's

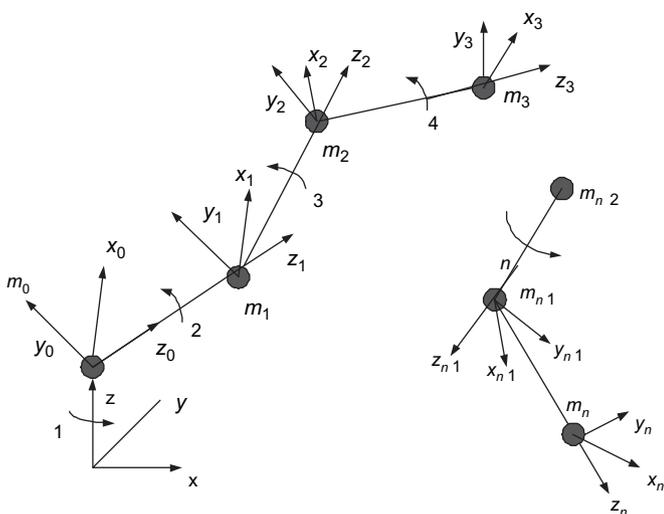


Fig. 8. Torsion angles.

method [5] p. 68–69 can be used, or we can sum $f_E(g)$ over all values of g . Note that $f_E(g)$ is normalized as a probability density in this section, whereas $f(g)$ was a number density in previous sections of the paper. $f_E(g)$ by itself is neither a true number density, nor is it a probability density, though it is an important quantity to compute along the path of finding $f_E(g)$. When there is no ambiguity, we will drop the subscript E .

Unfortunately, the calculation of f_E in Eq. (18) for the most general energy function, $E(\phi)$, is a problem requiring an exponential amount of computation in the chain length. In contrast, one of the simplest kinds of conformational energy function models only nearest neighbor interaction and leads to a separable partition function. In order for us to generate the statistical information needed to weight the relative occurrence of polymer conformations in a statistical mechanical ensemble we use the Rotational Isomeric State (RIS) model. Two basic assumptions to the RIS model have been developed by Flory:

- The conformational energy function for a chain molecule is dominated by interactions between each set of three groups of atoms at the intersection and ends of two adjacent bond vectors. Hence the conformational energy function can be written in the form $E(\phi) = \sum_{i=1}^{n-1} E_i(\phi_{i-1}, \phi_i)$, where $\phi = (\phi_0, \phi_1, \dots, \phi_{n-1})$ is the set of torsion angles.
- The value of the conformational weighting function $\exp(-E(\phi)/k_B T)$ is negligible except at the finite number of points where $E(\phi)$ is minimized. For our case we do not have to find the local minima since we are referring to lattice.

If we include only the effects of pairwise conformational energy, then the distribution function describing the end positions and orientations of a chain can be written in the form:

$$f(g) = \sum_{\phi \in I(g)} \exp\left(\frac{-\sum_{i=1}^{n-1} E(\phi_i, \phi_{i+1})}{k_B T}\right).$$

This is a form that is amenable to computation using a modified version of the space-group convolution method presented earlier. After performing these convolutions we can divide by Z to obtain a probability density.

The generalized lattice motion group convolution technique can be made to conform with the RIS assumptions and provides a numerical tool to generate the statistical distributions of interest. One can write the conformational energy function as $E(\phi) = \sum_{i=0}^{n-1} E_i(\phi_i, \phi_{i+1})$. The frame distributions for the lower and upper segments are generated as before. These frame distributions are not only function on G , but also depend on the bond angles contributing to the energy of interaction between the two chain segments. The brute-force method for finding frame distribution function requires initially to define an energy matrix 3×3 for the case of the tetrahedral lattice and for 4×4 the case of the cubic one. The entries of this matrix are given by $\exp(-E(\phi_i, \phi_{i+1})/k_B T)$, where ϕ_i and ϕ_{i+1} are the pairwise sequential torsion angles of the chain depicted in Fig. 8.

In the case of a chain with sequentially pairwise interactions, Eq. (7) can be written as:

$$f_{n_1+n_2}(g, \phi_0, \phi_n) = \sum_{\phi_i} \sum_{\phi_{i+1}} f_{n_1}(g, \phi_0, \phi_i) * f_{n_2}(g, \phi_{i+1}, \phi_n) e^{\frac{-E(\phi_i, \phi_{i+1})}{k_B T}} \quad (20)$$

Which can be rewritten more explicitly as:

$$f_{n_1+n_2}(g, \phi_0, \phi_n) = \sum_{j=0}^{z-1} \sum_{k=0}^{z-1} \sum_{h \in G} f_{n_1}(h, \phi_0, \phi_j^i) f_{n_2} \times (h^{-1} \circ g, \phi_{i+1}^k, \phi_n) e^{\frac{-E(\phi_i^j, \phi_{i+1}^k)}{k_B T}}. \quad (21)$$

The statistical weight matrix, W , is defined to have entries of the form:

$$w_{j,k} = \exp\left(\frac{-E(\phi_i^j, \phi_{i+1}^k)}{k_B T}\right)$$

where $j, k \in [0,1,2]$ for the tetrahedral case and $j, k \in [0,1,2,3]$ in the cubic case. For our numerical studies the entries of these matrices were chosen arbitrarily. The only constraint in this choice is that the matrix should be symmetric and have positive entries.

For the case of the tetrahedral lattice the matrix is of the form:

$$\begin{bmatrix} 0.8 & 0.2 & 0.3 \\ 0.2 & 0.5 & 0.4 \\ 0.3 & 0.4 & 0.7 \end{bmatrix} \quad (22)$$

whereas for the cubic lattice the matrix is:

$$\begin{bmatrix} 0.3 & 0.04 & 0.25 & 0.09 \\ 0.04 & 0.2 & 0.75 & 0.01 \\ 0.25 & 0.75 & 0.7 & 0.8 \\ 0.09 & 0.01 & 0.8 & 0.9 \end{bmatrix}. \quad (23)$$

For the cubic lattice ϕ_j, ϕ_i take four values: 0, $\pi/2$, π , $3\pi/2$, whereas for the tetrahedral lattice they take the three values: ϕ_j, ϕ_i : 0, $2\pi/3$, $4\pi/3$.

The definition of the convolution now incorporating the effects of the conformations are more complicated since the frame distribution functions have as arguments not only the positions and orientations but also the pair of the first and last torsion angles in the chain. Since each of the torsion angles takes 3 and 4 values for the tetrahedral and the square lattice, respectively, then the size of the array describing the function will be 9 and 16 times the size without incorporating the conformational effects, respectively. The definition of the convolution for this case is a modified version of Eq. (21). The values of the torsion angles for the cubic lattice are 0, 1, 2, 3. So for each pair of angles ϕ_0, ϕ_1 of the $f_{n_1+n_2}$ function we have 9 or 16 convolutions for the tetrahedral or the square lattice. Hence overall we have 81 or 256 space-group convolutions. Hence the time that it takes to compute all the possible conformations of the ensemble of the n -segment random walk on the tetrahedral or on the square lattice is 81 or 256 times longer when incorporating the conformational energy effects than when not.

5. Numerical results

In this section the outcome of our work is presented. By applying the space-group convolution method we obtain in very short time the pose probabilities for the distal ends of all of the possible n -segment torsional random walks on each lattice. What this means is that the ensemble properties of the polymer chain approximated by lattice models can be described more easily and faster. After first neglecting excluded volumes and assuming that all conformations have the same energy we obtain initial results. Then we proceed by assuming an excluded volume in the shape of circle or sphere (depending on whether the lattice is planar or spatial) and after obtaining these results we assume the case where each conformation has different energies (when taking into account the effects of different pairwise torsion angles).

It is important to mention that we validated the results for all the cases. For example, for the case where we do not have obstacles or energy effects we compared for a small number of segments the results of the convolution with the results of brute-force enumeration. In all cases we have an exact match. For larger number of segments when we cannot compare with brute-force results, the condition that $\sum_{g \in G} f_{n_i}(g) = z^n$ for n being the number of segments of the torsional random walk was checked for the case of no obstacles or pairwise interactions. This is a necessary condition for the method to be correct. For the case that we have either obstacles or torsion angles effects or both, we compared, for small number of segments, the lattice convolution results with the brute-force results and we had a complete match.

Once the pose probability distribution is found, it is easy to find distributions of end-to-end distance. We first marginalize (sum) over orientations to obtain a distribution in end-to-end positions. Then the distance of each lattice point from the

origin is recorded together with its corresponding number density as an ordered pair (distance, density). Each time a new lattice point is encountered with a distance from the origin that has been recorded previously, this density is added to the previous value. The resulting array is sorted by distance from smallest to largest. Bins of size 2 (apart from the case of the hexagonal lattice which is a size of 5) bond lengths are used to define histograms, and density values corresponding to distances that fall within each bin are added.

5.1. Square lattices

The next step is to present the results. For the case of the square lattice for $n = 50, 100, 150, 200$ with and without taking into account the obstacle. The plot on Fig. 9 presents the distribution of the *Euclidean* distances in the square lattice by taking a bin size of 2. We count all the walks that end in a distance greater or equal to 0 and less than 2 and we assign to them to the zero distance. The same procedure holds for $m = 2, \dots, n$, that is we count all the walks that end in a distance greater or equal to $m - 2$ and less than m and we assign to them to the $m - 2$ distance. The first four graphs refer to the no-obstacle case, whereas the rest refer to the obstacle case. It is worth noting, how much the obstacle affects the distribution. The percentage is taken over all possible 2^n walks. The results refer to (a) the distribution of 50 segments without obstacle, (b) 100 segments without obstacle, (c) 150 segments without obstacle, (d) 200 segments without obstacle, (e) 50 segments with obstacle, (f) 100 segments with obstacle, (g) 150 segments with obstacle, (h) 200 segments with obstacle.

On the other hand, the percentage of times that the same value of end-to-end *lattice* distance is reached for each case is shown in Fig. 10. The results refer to: (a) the distribution of 50 segments without obstacle, (b) 100 segments without obstacle, (c) 150 segments without obstacle, (d) 200 segments without obstacle, (e) 50 segments with obstacle, (f) 100 segments with obstacle, (g) 150 segments with obstacle, (h) 200 segments with obstacle.

It is interesting to observe how the ratio of number of torsion walks generated by obstacle over the number of walks generated without obstacle evolves as the number of segment increases, see Fig. 11.

Table 1 shows us the mean-square end-to-end distances for various number of segments of square lattice random walks with and without taking the specific obstacle into account.

That is given by the formula $\langle r^2 \rangle = \sum_{i=1}^n d_i r_i^2 / \sum_{i=1}^n d_i$, where d_i is the number of times each distance is reached while r_i is the actual Euclidean distance. It is worth mentioning that the mean-square end-to-end distance for the 1023 segments of non-reversal random walks for the square lattice is $\langle r^2 \rangle = 1023$.

5.2. Hexagonal lattices

In the case of the hexagonal lattice we present the analogous results for the same number of segments with and without taking into account the same obstacle as in the case of the

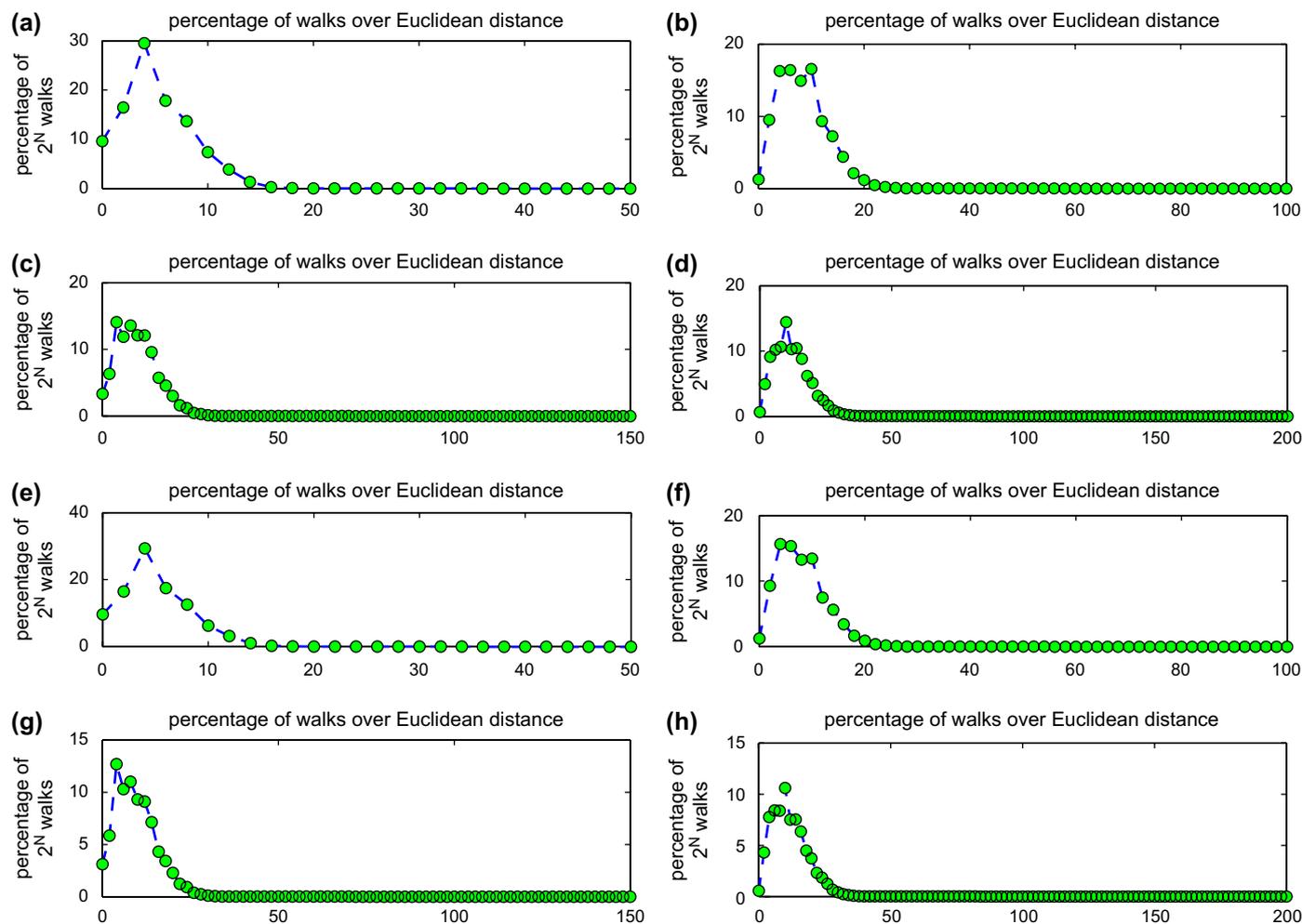


Fig. 9. Distribution of Euclidean distances for the square lattice.

square lattice. The plot in Fig. 12 presents the distribution of the Euclidean distances in the hexagonal lattice by taking a bin size of 2. So for distances from 0 to 2 we have the value at the 0 point, etc. The first four graphs refer to the no-obstacle case, whereas the rest refer to the obstacle case. It is worth noticing how much the obstacle affects the distribution. The percentage is taken over all possible 2^n walks. The results refer to (a) the distribution of 50 segments without obstacle, (b) 100 segments without obstacle, (c) 150 segments without obstacle, (d) 200 segments without obstacle, (e) 50 segments with obstacle, (f) 100 segments with obstacle, (g) 150 segments with obstacle, (h) 200 segments with obstacle.

Table 2 shows us the mean-square end-to-end distances for various number of segments of hexagonal lattice random walks with and without taking the specific obstacle into account.

5.3. Cubic lattices

Now in the case of the spatial lattices not only do we have higher computational complexity due to the additional dimension but we also have to consider whether all the conformations have the same or different energy. The latter is much more cost effective computationally than the former for

reasons that we described previously. Hence for the case when assuming all the conformations have the same energy we present results for $n = 50, 100$ segments (with and without obstacles) while for the case when we assume different energy states for different values of torsion angles we reduce the number of segments to 30. Presented in the Fig. 13 are the following: (a) the distribution of 50 segments without obstacle, (b) 100 segments without obstacle, (c) 50 segments with obstacle, (d) 100 segments with obstacle, (e) 30 segments without obstacle accounting for torsion angles (f) 30 segments with obstacle accounting for torsion angles. The mean-square end-to-end distances are presented on Table 3.

It is worth mentioning that in the case of the obstacle the number of random walks reduces to 1.2215×10^{30} from 1.2677×10^{30} for the 50-segments case, while for the 100 segments case it reduces to 3.6472×10^{56} from 1.6069×10^{60} .

5.4. Tetrahedral lattices

The results in Fig. 14 are: (a) $n = 50$ without obstacles, (b) for $n = 50$ with obstacles, (c) for $n = 30$ without obstacle for the case where different energy states have different weight,

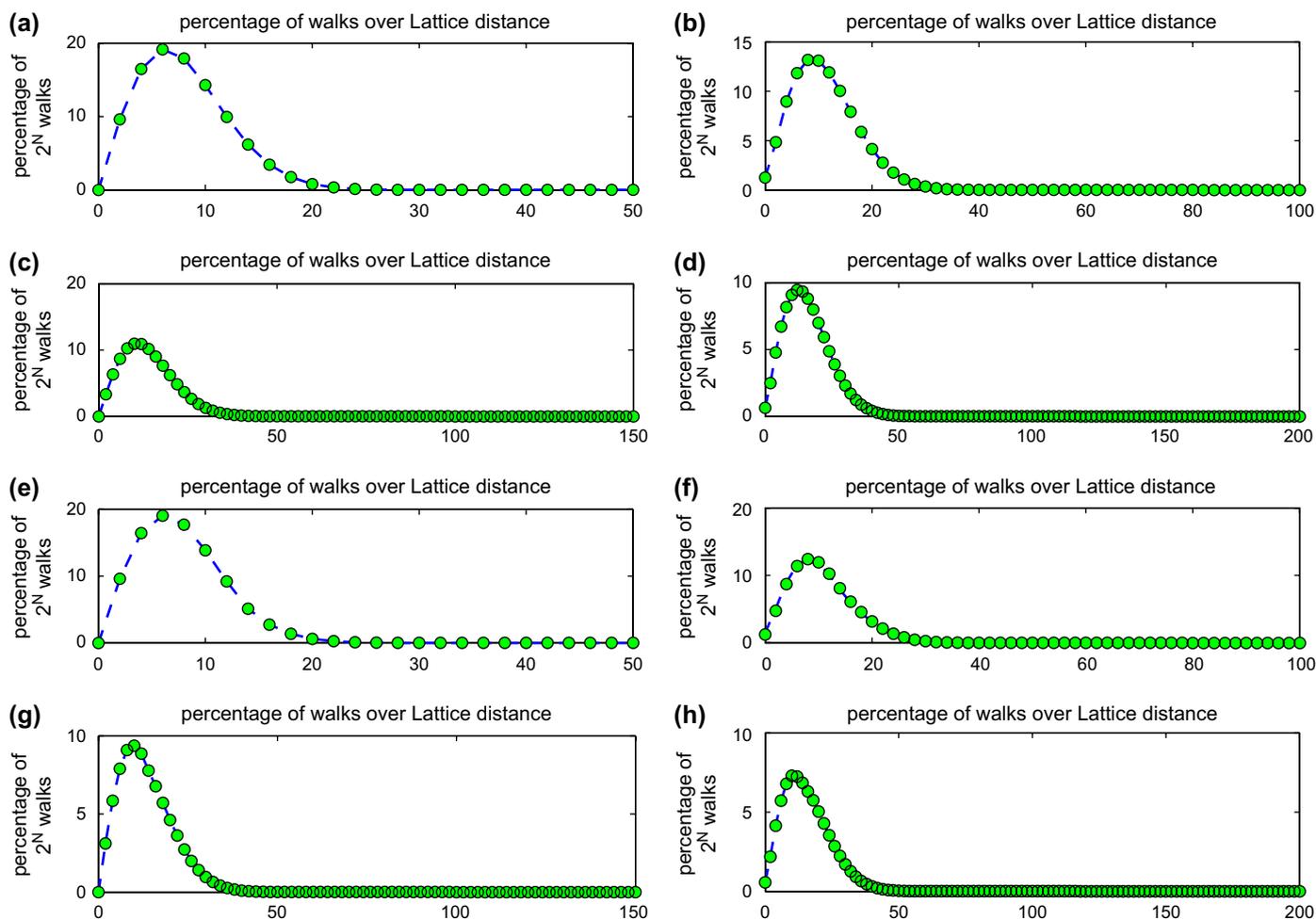


Fig. 10. Distribution of lattice distances for the square lattice.

(d) for $n = 30$ with obstacle for the case where different energy states have different weight. The mean-square end-to-end distances are presented in Table 4.

It is worth mentioning that in the case of the obstacle the number of random walks reduces to 7.1630×10^{23} from 7.1790×10^{23} for the 50-segments case.

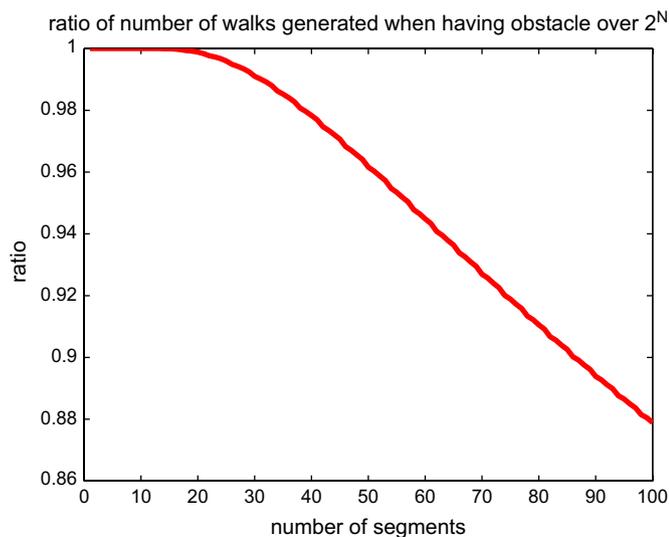


Fig. 11. Ratio of number of walks when having obstacle over the number of walks when not having obstacle.

6. Cost for making these calculations

Since the low computational cost is one of the main benefits of this new method, we discuss it separately here. For the square lattice, computing the distribution functions corresponding to all conformations of all torsional walks of length up to 200 is done in 75 s on a PC with Pentium 4 processor (3.2 GHz), 1 GB RAM with Red Hat Linux as the operating system. Computing the distribution functions for all walks

Table 1
Mean-square end-to-end distance – square lattice

Number of links/with or without obstacles, no energy state effect	Without obstacles	With obstacles
50	50.0	47.2629
100	100.0	92.8864
150	150.0	141.7215
200	200.0	193.2174

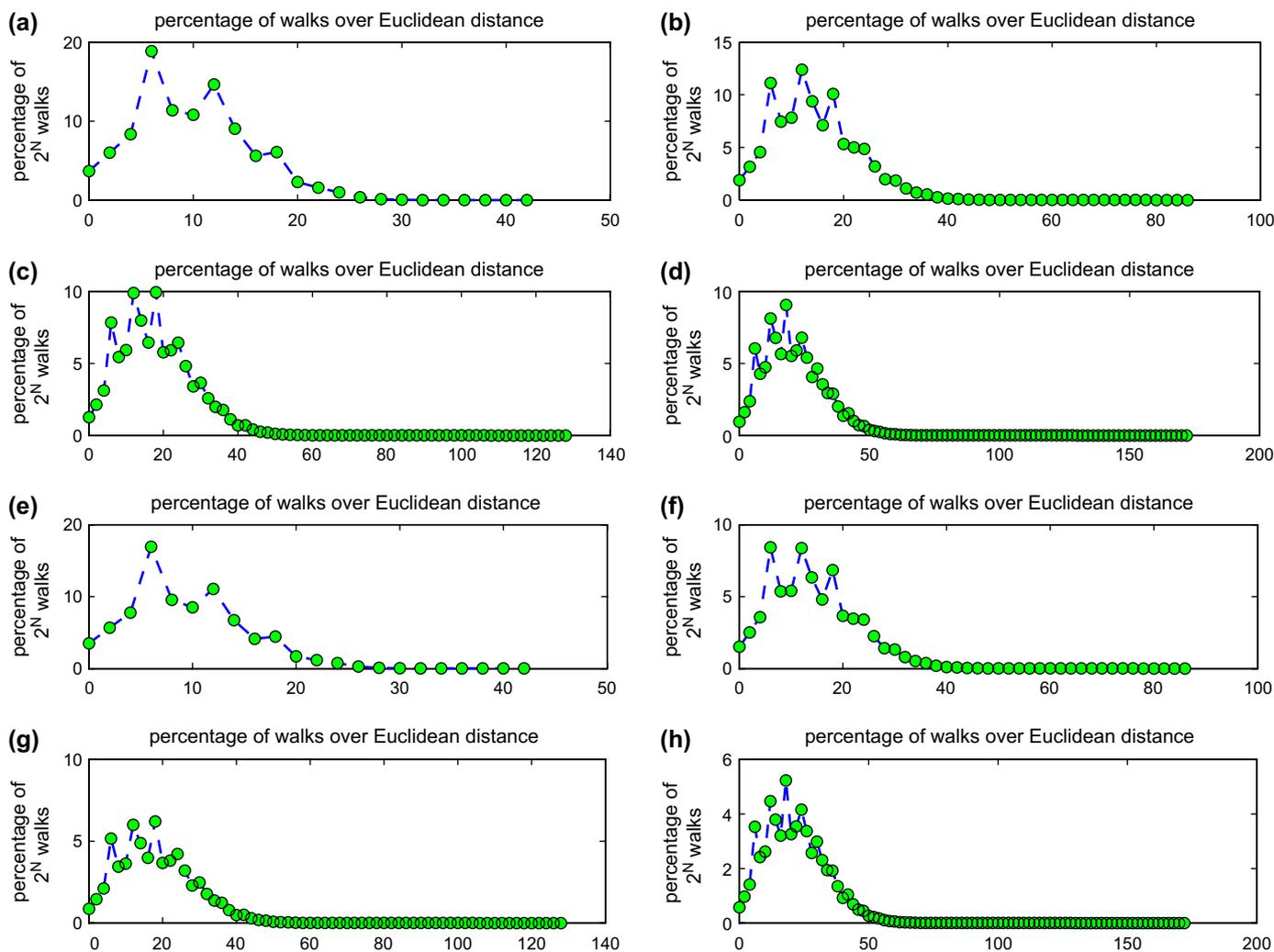


Fig. 12. Distribution of Euclidean distances for the hexagonal lattice.

up to 1023 segments takes 10 000 s. For the hexagonal lattice we have the issue that in our calculation we introduce square roots which slows down the calculations. Even so, the distribution functions for all walks up to length 200 segments are computed in 6000 s, or less than 2 h. For the case of the spatial lattices without considering the effects of torsion angles the calculations are much slower than the planar case. In the cubic lattice, computing the distribution functions for all walks up to length 100 takes 40 000 s, almost 12 h. For the tetrahedral case, the distribution functions for all walks of length up to

50 take 7330 s. For the case when we take into account the torsion angles, that time/cost is multiplied by 256 and 81, respectively.

7. Comparison with classical results

In this section we demonstrate that our convolution method produces the same results as classical formulations for the special case of chains without sequentially proximal interactions or obstacles. In particular, we consider the relative probability of the most outstretched conformations, ring closure probabilities, and moments of both end-to-end distance and radius of gyration.

For the specific case of the non-reversal walks of the cubic lattice the normalized probability that the most distal points are reached can be rewritten as $5/5^n$. This means that for each n -non-reversal walk the maximum distance is reached only by 5 walks. That is true and verified. In addition to that, the maximum distance value should always be n (since $l = 1$ for the cubic lattice case). This has also been verified in our data.

Table 2
Mean-square end-to-end distance – hexagonal lattice

Number of links/without obstacles	
50	146.00
100	296.00
150	446.00
200	596.00
Number of links/with obstacles	
50	111.3886
100	206.3397
150	295.6008
200	380.5064

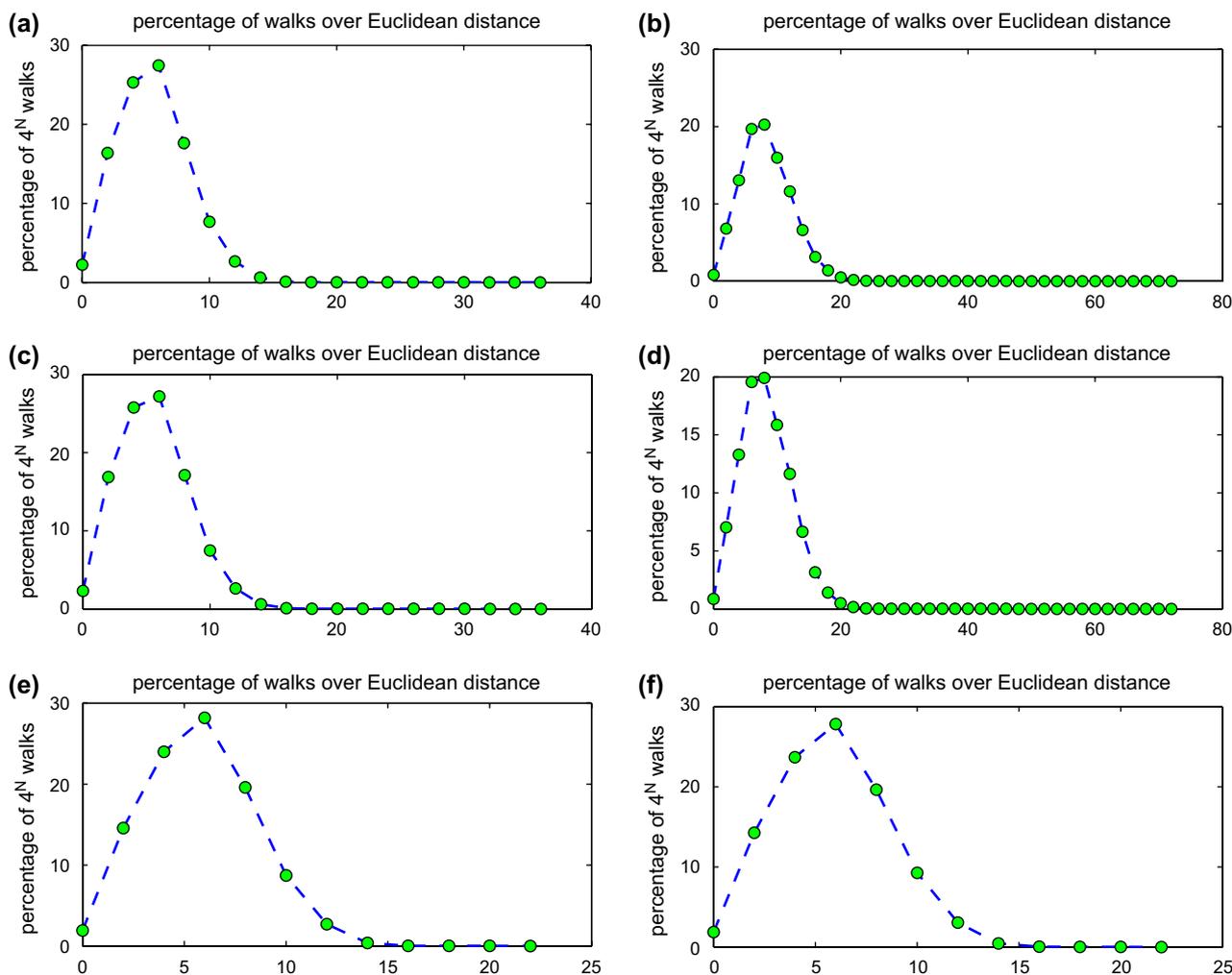


Fig. 13. Distribution of Euclidean distances for the cubic lattice.

We also compared our formulation with the Zimm result [44], and the Jacobson–Stockmayer result [43]. Direct comparison shows asymptotic convergence to those results. If we modify the Zimm formulation, which was derived for the case of a freely jointed chain, by taking averages over the accessible states in the lattice rather than a uniform spherical average, our results match exactly with this modified Zimm formulation for polymers of all lengths. The Jacobson–Stockmayer result is based on the assumption of a Gaussian chain, which is also valid only asymptotically.

Table 3
Mean-square end-to-end distance – cubic lattice

Number of links/without obstacles neglecting energy states	Without obstacles
50	50.00
100	100.00
Number of links/with obstacles neglecting energy states	
50	47.4649
100	87.1115
Number of links/without obstacles different energy states	
30	51.999
Number of links/with obstacles different energy states	
30	51.2488

The details of how to calculate classical quantities of interest using our methodology, and comparison with the classical results are presented in the following subsections.

7.1. Comparison with the Jacobson–Stockmayer result

Concerning the Jacobson–Stockmayer result [43], we have the fraction of chain configurations permitting ring closure that is given by the formula:

$$P = \left(\frac{3}{2\pi\nu n} \right)^{3/2} \frac{u_s}{b^3} \tag{24}$$

Eq. (24) gives the probability for the conformations with $r = 0$. In Eq. (24) the quantity u_s is a small volume that other atoms occupy if one atom is fixed when a ring is formed. The quantity b is the effective link length. Since we are dealing with lattices, the effective link length is always a constant and the volume u_s that we referred to above is constant too, and we can say that $u_s = b^3$ thus $(u_s/b^3) = 1$. Furthermore, ν in Eq. (24) is the number of chain atoms in monomer units which in our case (since we are dealing with lattice random walks

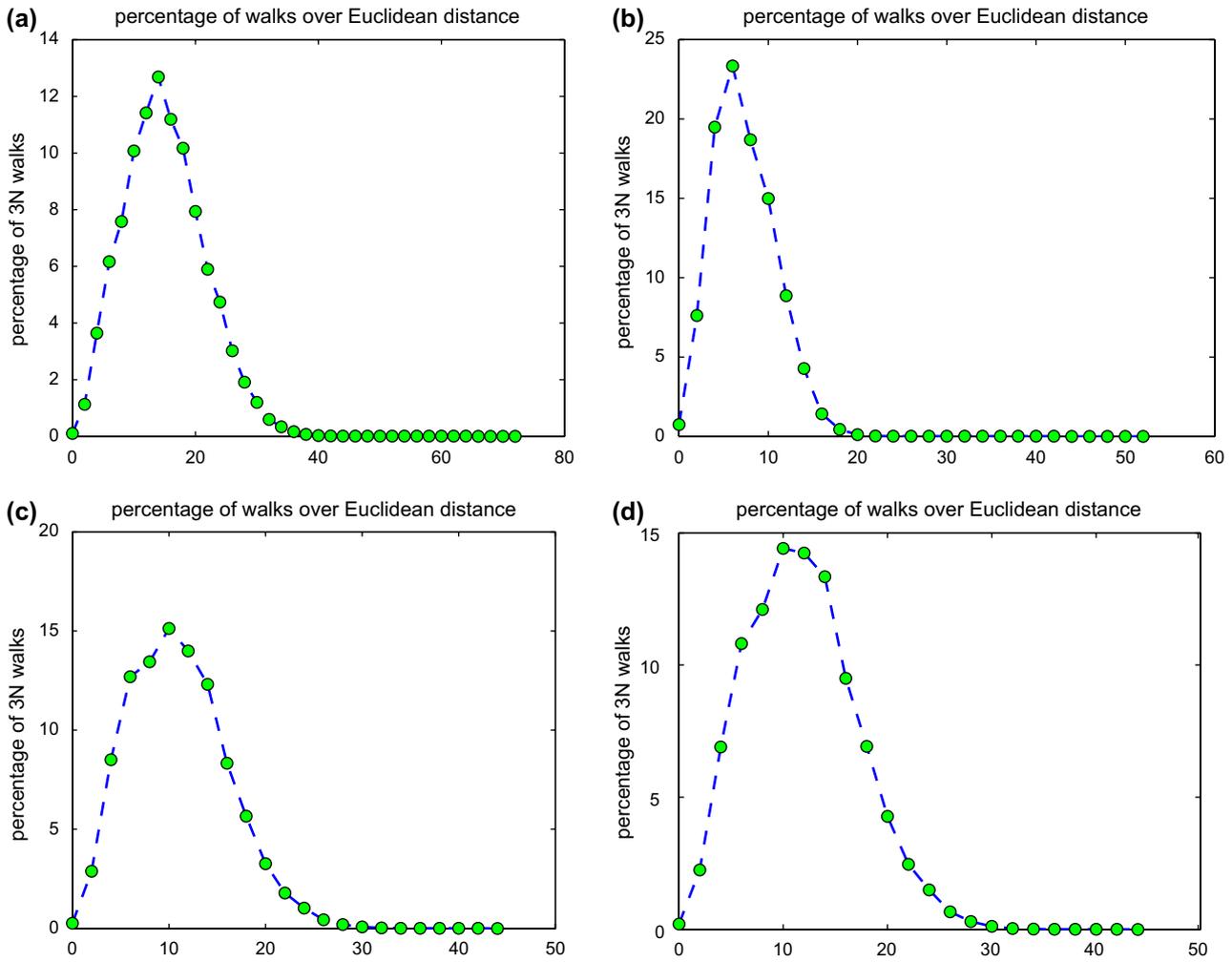


Fig. 14. Distribution of Euclidean distances for the tetrahedral lattice.

Number of links/without obstacles neglecting energy states	Without obstacles
50	295.5000
Number of links/with obstacles neglecting energy states	
50	78.7260
Number of links/without obstacles different energy states	
30	166.8206
Number of links/with obstacles different energy states	
30	187.9892

and we count each lattice site as one unit) we take $\nu = 1$. Therefore, Eq. (24) is reduced to:

$$P = \left(\frac{3}{2\pi n} \right)^{3/2} \quad (25)$$

In Fig. 15 we present the probability of $r = 0$ for each value of n for both the method of the Jacobson–Stockmayer and the convolution for the tetrahedral lattice. The thicker line

represents the Jacobson–Stockmayer value. As we see initially there is a small divergence but as n increases the difference becomes indistinguishable. Thus the probability of ring closure for the tetrahedral lattice follows the Jacobson–Stockmayer result. The same holds for the cubic lattice.

7.2. Comparison with the Zimm result

According to Ref. [44] the mean-square radius of gyration of chains with $r = 0$ is $R^2 = b^2N/12$, where R^2 is the mean-square radius of gyration of the rings and b^2N is the mean-square end-to-end distance. That is the average mean-square end-to-end distance of an n -segment random walk is 12 times the mean-square radius of gyration of the rings. In our graph we show that for the tetrahedral lattice the ratio (mean-square end-to-end distance over mean-square radius of gyration of rings, that is, b^2N/R^2) approaches the number 12.

It is well known [5,36,39] that the mean-square radius of gyration is given by:

$$\langle s^2 \rangle = \frac{1}{2(n+1)^2} \sum_{i=0}^{n-1} \sum_{j=i+1}^n \langle x_{ij}^2 \rangle$$

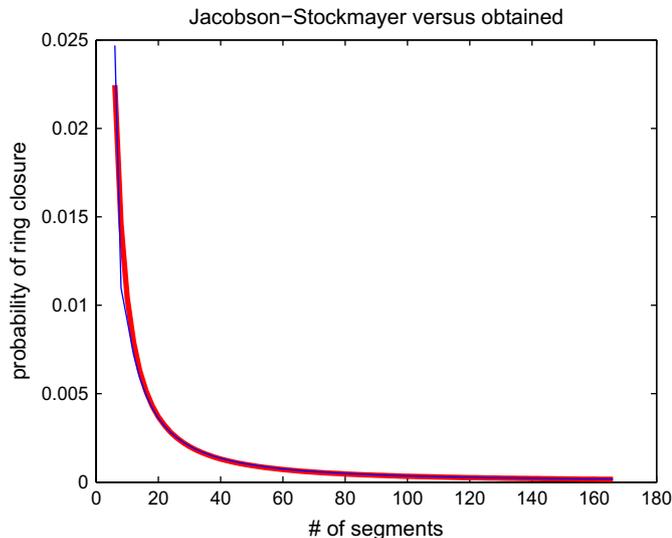


Fig. 15. Result of Jacobson–Stockmayer versus the one obtain by convolution.

where $\langle x_{ij}^2 \rangle$ is the mean-square end-to-end distance from i th segment to j th one. This means that we need to find the number density that bead j will reach a particular pose relative to bead i given constraints on the proximal and distal ends. In ring closure, the distal and the proximal are fixed to the origin. For a fixed distal end pose g we have that $f_n(g) = (f_i^* f_{j-i}^* f_{n-j})(g)$. When we fix the distal end at $g = e$, ring closure results. In this case, the probability/number density that bead j reaches pose z relative to the reference frame at bead i is:

$$F_e^{i,j}(z) = \frac{f_{j-i}(z)}{f_n(e)} \sum_{h \in G} f_i(h) f_{n-j}(z^{-1} \circ h).$$

By finding the mean-square end-to-end distance associated with $F_e^{i,j}(z)$ (by marginalizing over all variables except for the radial direction) we can find $\langle x_{ij}^2 \rangle$ in the expression for the radius of gyration of the rings. The details for the case of the tetrahedral lattice become slightly more complicated due to its ‘two definability’, but the basic principle is the same.

In Fig. 16 we see the mean-square radius of gyration of rings with respect the number of segments for the tetrahedral lattice (we have rings only for even number of segments) in the upper plot, whereas in the lower plot we see the ratio $b^2 N / R^2$. We see that this ratio approaches 12.

7.3. Shape of the distribution between the limits

The shape of the distribution function for r is specified by dimensionless ratios constructed from the even moments, $\langle r^{2p} \rangle_0 / \langle r^2 \rangle_0^p$. The ratio with $p = 2$ specifies the width, the ratio with $p = 3$ specifies the skewness, etc. In the limit, as $n \rightarrow \infty$, these ratios are known precisely (as ratios of positive integers) for many values of p . Indeed the importance of these properties is stated also in Ref. [6] in chapter 2 written by Akten,

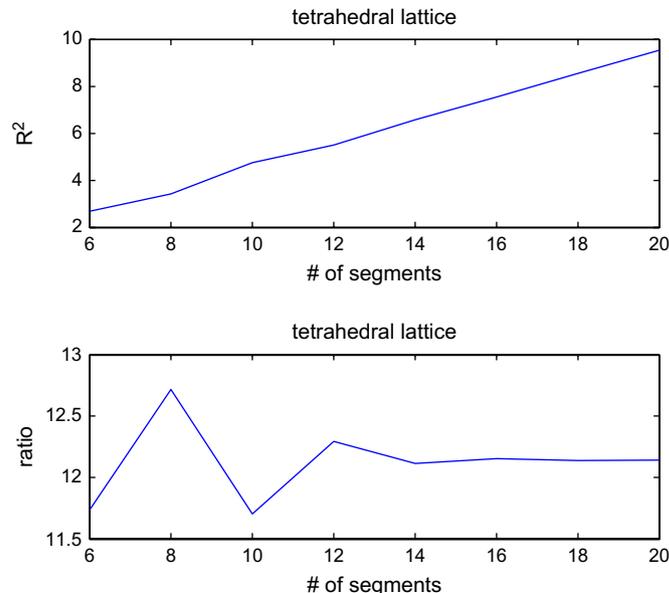


Fig. 16. Verification of the Zimm result for the tetrahedral lattice.

Mattice and Suter. The asymptotic values of these ratios are given precisely in Ref. [36] and these are:

$$\frac{\langle r^4 \rangle_0}{\langle r^2 \rangle_0^2} = \frac{5}{3}, \frac{\langle r^6 \rangle_0}{\langle r^2 \rangle_0^3} = \frac{35}{9}, \frac{\langle r^8 \rangle_0}{\langle r^2 \rangle_0^4} = \frac{35}{3}$$

This asymptotic value of the ratios of the even moments of the mean end-to-end distance was initially calculated by Flory and Jernigan [5,46,48,47]. The most computationally convenient method for calculating both the mean radius of gyration and its higher moments and the even moments of the mean end-to-end distance is presented in a work by Mattice and Sienicki [45]. Before trying to calculate the even moments of the mean end-to-end distance by using the convolution method we calculated both the even moments of the mean end-to-end distance and the even moments of the mean radius of gyration by using the method presented in that paper [45]. The way we did this was for the cubic lattice to set in Eq. (8) of [45] $\theta = \pi/2$ and $\phi = 0, \pi/2, \pi, 3\pi/2$, and for the tetrahedral lattice $\theta = 1.2310$ and $\phi = 0, 2\pi/3, 4\pi/3$. For the lengths l we set for the cubic lattice the number one and for the tetrahedral lattice the number $\sqrt{3}$. After working according to the paper we obtain the values of these moments with respect to the number of segments. Next we proceeded by calculating the ratios dividing accordingly. The results (obtained according to the method of Mattice and Sienicki [45]) converged to the specific value. Next we calculated through the convolution method the values of the even moments of the mean end-to-end distance for the tetrahedral lattice. The method for calculating these quantities is similar to the calculation of the second moment (mean-square end-to-end distance). We compare these results with the one obtained by the method of Mattice and we had absolute match.

In Figs. 17–19 we present the comparison and the convergence of the ratios of the 4th, 6th and 8th moment of the

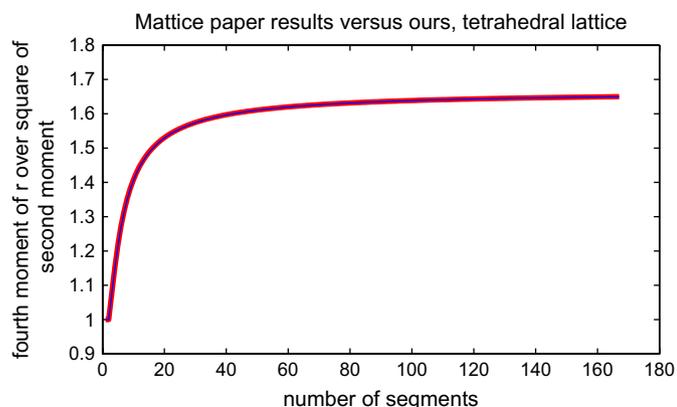


Fig. 17. Ratio4 for the tetrahedral lattice obtain both by the convolution and the Mattice method.

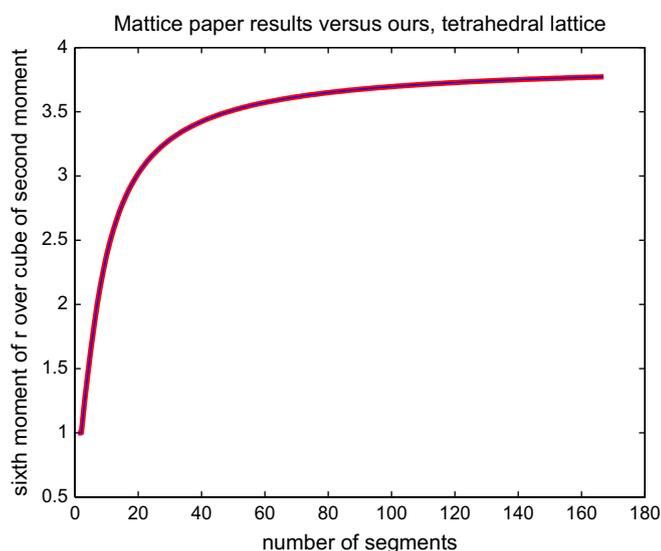


Fig. 18. Ratio6 for the tetrahedral lattice obtain both by the convolution and the Mattice method.

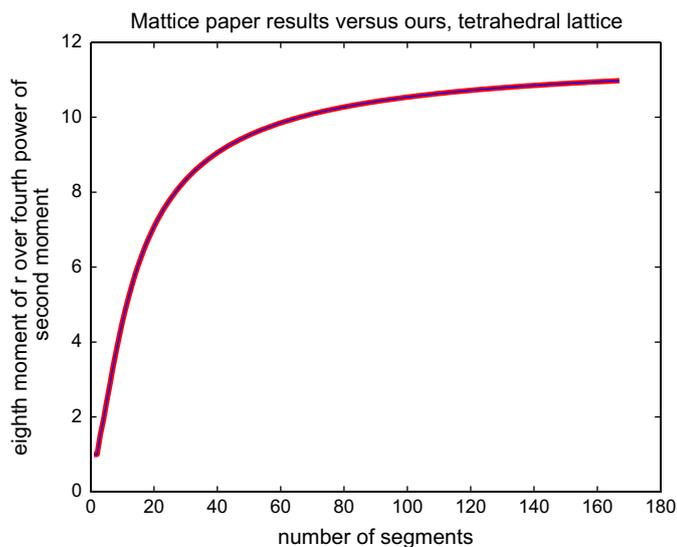


Fig. 19. Ratio8 for the tetrahedral lattice obtain both by the convolution and the Mattice method.

end-to-end distance for the tetrahedral lattice. The thick line corresponds to the result obtained by the Mattice method. As we see the two methods give the same results and converge to the desired values. It is worth mentioning that for finding these ratios Mattice method is much faster.

8. Conclusions

This paper presented a new method for generating the conformational statistics of lattice models of chain molecules. Poses of reference frames attached to the chain as it intersects with each of the lattice sites served as inputs. All of the distribution functions corresponding to the z^n independent lattice conformations were generated for all values of n up to n in $O(n^{D+1})$ arithmetic operations for $D = 2$ or 3 dimensional lattices. That was achieved by performing convolutions with respect to the crystallographic space group for each lattice. The formulation was modified to include the effects of an obstacle (excluded volume) in a lattice, and to calculate the frequency of the occurrence of each conformation when the effects of pairwise conformational energy are included. In the latter case (which was for three dimensional lattices only) the computational cost is $O(z^4 n^3)$. This polynomial complexity is a vast improvement over the exponential $O(z^n)$ complexity associated with the brute-force enumeration. The distribution of end-to-end distances and average radius of gyration were calculated. The method was demonstrated on square, hexagonal, cubic and tetrahedral lattices. The next step of this work is to extend our research on finding tight upper and lower bounds on self-avoiding walks.

Acknowledgement

This work was supported under NIH grant R01 GM075310 ‘Group Theoretic Methods in Protein Structure Determination’.

References

- [1] Chirikjian GS. Conformational statistics of macromolecules using generalized convolution. *Computational and Theoretical Polymer Science* February 2001;11:143–53.
- [2] Chirikjian GS. Fredholm integral equations on the Euclidean motion group. *Inverse Problems* 1996;12:579–99.
- [3] Chirikjian GS, Ebert-Uphoff I. Numerical convolution on the Euclidean group with applications to workspace generation. *IEEE Transactions on Robotics and Automation* 1998;14(1):123–36.
- [4] Rosenbluth MN, Rosenbluth AW. Monte-Carlo calculation of the average extension of molecular chains. *Journal of Chemical Physics* 1955;23(2):356–9.
- [5] Flory PJ. *Statistical mechanics of chain molecules*. Cincinnati, OH: Hanser/Gardner Publications; 1985.
- [6] Kotlyanskii M, Theodorou DN. *Simulation methods for polymers*. New York, NY: Marcel Dekker Inc.; 2004.
- [7] Kuhn W. Concerning the shape of thread shapes molecules in solution. *Kolloid-Zeitschrift* 1934;68:2–15.
- [8] Alexandrowicz Z. Monte Carlo of chains with excluded volume – a way to evade sample attrition. *Journal of Chemical Physics* 1969;51:561–5.

- [9] Collet O, Premilat S. Calculation of conformational free-energy and entropy of chain molecules with long-range interactions. *Macromolecules* 1993;26:6076–80.
- [10] He SQ, Scheraga HA. Macromolecular conformational dynamics in torsional angle space. *Journal of Chemical Physics* 1998;108:271–86.
- [11] Honeycutt JD. A general simulation method for computing conformational properties of single polymer chains. *Computational and Theoretical Polymer Science* 1998;8:1–8. Part 2.
- [12] Hiller A, Wall FT, Wheeler DJ. Statistical computation of mean dimensions of macromolecules. *Journal of Chemical Physics* 1954;22:1036.
- [13] Wall WA, Seitz FT. Simulation of polymers by self-avoiding nonintersecting random chains at various concentrations. *Journal of Chemical Physics* 1977;67:3722–6.
- [14] Alexandrowicz Z, Accad Y. Monte Carlo of chains with excluded volume: distribution of intersegmental distances. *Journal of Chemical Physics* 1971;54:5338–45.
- [15] Khokhlov AR, Grosberg AY. *Statistical physics of macromolecules*. New York, NY: ATP Press; 1994.
- [16] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of state calculations by fast computing machines. *Journal of Chemical Physics* 1953;21:1087–92.
- [17] Kremer K, Binder K. Monte Carlo simulations of lattice models for macromolecules. *Computer Physics Reports* 1988;7:259.
- [18] Binder K, Heerman DW. *Monte Carlo simulations in statistical physics: an introduction*. Berlin, Germany: Springer; 1997.
- [19] Milchev A, Binder K, Heerman DW. Fluctuations and lack of self-averaging in the kinetics of domain growth. *Zeitschrift für Physik B: Condensed Matter* 1986;63:521–35.
- [20] Sokal AD. *Monte Carlo methods for the self-avoiding walk, Monte Carlo and molecular dynamics simulations in polymer science*. New York, NY: Oxford University Press; 1995.
- [21] Flory PJ. *Principles of polymer chemistry*. Ithaca, NY: Cornell University Press; 1953.
- [22] Sariban A, Binder K. Phase-separation of polymer mixtures in the presence of solvent macromolecules. *Macromolecules* 1988;21:711–26.
- [23] Gibbs JH, Di Marzio EA. Nature of the glass transition and the glassy state. *Journal of Chemical Physics* 1958;28:373–83.
- [24] Di Marzio EA, Gibbs JH. Chain stiffness and the lattice theory of polymer phases. *Journal of Chemical Physics* 1958;28:807–13.
- [25] Wolfgardt M, Baschnagel J, Paul W, Binder K. Entropy of glassy polymer melts: comparison between Gibbs-Di Marzio theory and simulation. *Physical Review E* August 1996;54(2):1535–43.
- [26] Binder K, Gausterer H, Lang CB. *Computational methods in field theory*. Berlin, New York: Springer; 1992.
- [27] Carmesin I, Kremer K. The bond fluctuation method – a new effective algorithm for the dynamics of polymers in all spatial dimensions. *Macromolecules* 1988;21:2819–23.
- [28] Deutsch HP, Binder K. Interdiffusion and self-diffusion in polymer mixtures – a Monte-Carlo study. *Journal of Chemical Physics* 1991;94:2294–304.
- [29] Weber H, Paul W, Binder K. Monte Carlo simulation of a lyotropic first-order isotropic–nematic phase transition in a lattice polymer model. *Physical Review E* 1999;59(2):2168–74.
- [30] Muller M, Binder K, Oed W. Structural and thermodynamic properties of interfaces between coexisting phases in polymer blends – a Monte-Carlo simulation. *Journal of the Chemical Society, Faraday Transactions* 1995;91:2369–79.
- [31] Wall FT, Mandel F. Macromolecular dimensions obtained by an efficient Monte-Carlo method without sample attrition. *Journal of Chemical Physics* 1975;63:4592–5.
- [32] Lal M. Monte Carlo computer simulation of chain molecules. *Molecular Physics* 1969;17:57.
- [33] Madras N, Sokal AD. The pivot algorithm – a highly efficient Monte-Carlo method for the self-avoiding walk. *Journal of Statistical Physics* 1988;50:109–86.
- [34] Olaj OF, Lantschbauer W. Simulation of chain arrangement in bulk polymer. 1. Chain dimensions and distribution of the end-to-end distance. *Makromolekulare Chemie, Rapid Communications* 1982;3:847–58.
- [35] Mansfield ML. Monte-Carlo studies of polymer-chain dimensions in the melt. *Journal of Chemical Physics* 1982;77:1554–9.
- [36] Mattice WL, Suter UW. *Conformational theory of large molecules: the rotational isomeric state model in macromolecular systems*. New York, NY: Wiley; 1994.
- [37] de Gennes PG. *Scaling concepts in polymer physics*. Ithaca, NY: Cornell University Press; 1979.
- [38] Kinzel W, Reents G. *Physics by computer*. Berlin, Germany: Springer; 1998.
- [39] Chirikjian GS, Kyatkin AB. *Engineering applications of noncommutative harmonic analysis: with emphasis on rotation and motion groups*. Boca Raton, FL: CRC Press; 2001.
- [40] Schuyler AD, Chirikjian GS, Lu JQ, Johnson HT. Random-walk statistics in moment-based O(N) tight binding and application in carbon nanotubes. Art. No. 046701 Part 2. *Physical Review E* 2005;71.
- [41] Mark JE, Abou-Hussein R, Sen TZ, Kloczkowski A. Some simulations on filler reinforcement in elastomers. *Polymer* 2005;46:8894–904.
- [42] Sharaf MA, Mark JE. Monte Carlo simulations on the effects of nanoparticles on chain deformations and reinforcement in amorphous polyethylene networks. *Polymer* 2004;45:3943–52.
- [43] Jacobson H, Stockmayer WH. Intramolecular reaction polycondensations. I. The theory of linear systems. *Journal of Chemical Physics* 1950;18(12):1600–6.
- [44] Zimm BH, Stockmayer WH. The dimensions of chain molecules containing branches and rings. *Journal of Chemical Physics* 1949;17(12):1301–14.
- [45] Mattice WL, Sienicki K. Extent of the correlation between the squared radius of gyration and squared end-to-end distance in random flight chains. *Journal of Chemical Physics* 1989;90(3):1956–9.
- [46] Flory PJ, Jernigan RL. Second and fourth moments of chain molecules. *Journal of Chemical Physics* 1965;42(10):3509–19.
- [47] Jernigan RL, Flory PJ. Moments of chain vectors for models of polymer chains. *Journal of Chemical Physics* 1969;50(10):4178–85.
- [48] Flory PJ. Foundations of rotational isomeric state theory and general methods for generating configurational averages. *Macromolecules* 1974;7(3):381–92.
- [49] <http://capsicum.me.utexas.edu/ChE386K/html/tetrahedron.htm>.
- [50] <http://www.hermetic.ch/compsci/lattgeom.htm>.