Position and Orientation Distributions for Non-Reversal Random Walks using Space-Group Fourier Transforms

Aris Skliros[1], **Wooram Park**[2], and **Gregory S. Chirikjian**[2],[*]

[1]Department of Biochemistry, Biophysics and Molecular Biology, Iowa State University, USA

[2]Department of Mechanical Engineering, Johns Hopkins University, USA

Abstract

This paper presents an efficient group-theoretic approach for computing the statistics of non-reversal random walks (NRRW) on lattices. These framed walks evolve on proper crystallographic space groups. In a previous paper we introduced a convolution method for computing the statistics of NRRWs in which the convolution product is defined relative to the space-group operation. Here we use the corresponding concept of the fast Fourier transform for functions on crystallographic space groups together with a non-Abelian version of the convolution theorem. We develop the theory behind this technique and present numerical results for two-dimensional and three-dimensional lattices (square, cubic and diamond). In order to verify our results, the statistics of the end-to-end distance and the probability of ring closure are calculated and compared with results obtained in the literature for the random walks for which closed-form expressions exist.

Keywords

1. Introduction

The topic of random walks on lattices and on finite groups has been of interest to mathematicians for at least half a century [15,24,36], and remains of interest in recent times [9,10]. Often the main mathematical goals in studying random walks center around limiting behaviors of ensembles of walks as their length becomes large [28,41,43]. Similar topics have been studied intensely in the context of random walks on Lie groups, and associated convolutions (see e.g., [6,16,19]). Statements about the behaviors of random walks on broad classes of groups are often sought. However, our goal is somewhat different. In this paper, we develop a *computational methodology* to compute *exact* probability distributions of random walks of finite (but possibly large) length on particular finite groups. From these distributions we can extract any statistical quantity of interest. In spirit, our motivation is similar to that behind the development of group FFT's Diaconis and Rockmore [8], though the groups of interest in our study have noncommutative subgroups that remain small as the group grows, and so the usual Abelian FFT can be used. In particular, our emphasis is on random walks on proper crystallographic space groups, which are discrete subgroups of the groups of rigid-body motions. This particular family of groups is important because they provide a natural setting

[*]Corresponding author. gregc@jhu.edu.
askliro1@iastate.edu (A. Skliros), wpark7@jhu.edu (W. Park)

to study *non-reversal* random walks in lattices in Euclidean space. Such walks serve as an intermediate between those that are purely random and those that are completely self-avoiding (such as those studied in [25,30]). Walks on lattices have been studied extensively for a variety of reasons [42].

1.1. Literature review

Many scientific fields use random and self-avoiding walks on lattices. Polymer theory is an important example of this. The statistical behavior of long polymer chains on lattices is often studied using the Monte Carlo sampling method introduced in [34]. This method has been used extensively in the literature [39]. The existence of polymer science traces back to the 1930s [27]. Through the decades of the 1950s and 60s many more new models were introduced, as reviewed in Flory's book [13]. Studying the statistics of polymers is a matter of interest [1, 18] because of the knowledge about mechanical properties that results [26]. Some examples of these properties include the elasticity of rubbery materials [26] and the toughness and ductility of semi crystalline polymers. Another very important quantity is the entropy of the polymer solution. All of these properties depend on the statistics that we study [13,26], which in turn are due to their molecular structure. Hence it is very important to study the structure and statistics of ensembles of long polymer molecules.

A polymer configuration can be represented as a lattice random walk if we consider the lattice bead the central carbon atom of the monomer and the lattice edge the covalent bond between the monomers. Let $\mathbf{r}_i$ denote the position of the ith bead, and let $i = 0, \ldots, n$ enumerate the $n + 1$ lattice beads. Then the bond vector, $\mathbf{b}_i = \mathbf{r}_i - \mathbf{r}_{i-1}$, connects two sequential lattice beads. Summing the n edge vectors of $n + 1$ identical monomers in the chain, we find the end-to-end distance vector $\mathbf{r}$ [26]:

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

(1)

The corresponding end-to-end distance is $r = \|\mathbf{r}\|$ where $\|\mathbf{r}\| = \sqrt{\mathbf{r} \cdot \mathbf{r}}$. As a matter of notation, double vertical lines denote the two-norm of a vector, whereas single vertical lines denote the number of elements in a set, e.g., $|\{a, b, c\}| = 3$. Both notations are used in this paper and should not be confused.

Other important physical quantities are related to the distribution of values of r, denoted as $p(r)$. A plethora of simulation methods, for instance those in [31], have been developed in order to obtain information about how $p(r)$ evolves for polymer chains. Our study focuses on the random walk (RW) on lattices under the simple constraint that step $i + 1$ cannot return to the location of the walk at step i. This is called a "non-reversal random walk" (NRRW). It is convenient to think of NRRWs as *framed* walks. That is, a reference frame is attached to each step in the walk, with the local *x*-axis pointing along the new direction of motion, which in the case of a polymer is the bond vector.

1.2. Limitations of brute-force enumeration

Let g_n denote the reference frame attached to the nth bead in a polymer model. This frame consists of a rotation-translation pair, $g_n = (R_n, \mathbf{t}_n)$, where R_n is a rotation matrix that is an element of a proper crystallographic point group and $\mathbf{t}_n$ is a translation vector within the corresponding crystallographic lattice. In the brute-force case if one wants to move (without backtracking) from the reference frame attached to the end step of an *n*-segment walk which is described by g_n to the reference frame of an $(n + 1)$-segment random walk, then one could proceed as follows. First find all the possible poses (positions and orientations) that the 1-

segment random walk on the lattice can reach using allowable moves. The number of these moves is 3 for the square lattice, 3 for the diamond lattice and 5 for the cubic lattice. If we want to find all the walks of n segments, the computational complexities of doing this brute-force enumeration are 3^n, 3^n, and 5^n respectively. However, if one does not seek to record all possible conformations, but rather is only interested in the distribution of positions and orientations of the distal end relative to the proximal end, then given the distribution $f_n(g)$ and the corresponding distribution of 1-step walks that do not backtrack, $f_1(g)$, the distribution for all conformations of length $n + 1$ is obtained by the convolution [40]

$$f_{n+1}(g) = (f_n * f_1)(g) = \sum_{h \in G} f_n(h) f_1\left(h^{-1} \circ g\right).$$

(2)

Here the sum is taken over the proper crystallographic motion group, G, that describes the particular lattice walk. But since in practice the support of $f_1(g)$ is very small, the sum does not need to be taken over this whole group, but only over the support of f_1 for each value of g in the support of f_{n+1}.[†] Referring to [40] for more details, (2) can be written more explicitly as

$$f_{n+1}(\mathbf{t}, R) = (f_n * f_1)(\mathbf{t}, R) = \sum_{A \in \mathbb{P}} \sum_{\mathbf{a} \in \mathbb{L}} f_n(\mathbf{a}, A) f_1\left(A^T(\mathbf{t} - \mathbf{a}), A^T R\right).$$

(3)

where $\mathbb{P}$ is the proper crystallographic point group (described as a set of rotation matrices) and $\mathbb{L}$ is the lattice translation group (described as a set of vectors), combinations of which comprise all of the possible motions in G. Iterating these convolutions by using the result of each as the input for the next and letting $n \rightarrow n + 1$ generates the sequence of distributions $\{f_n(g) | n = 1, 2, 3, \dots\}$ in polynomial complexity rather than the exponential complexity of brute-force enumeration. However, for very long chains, it is possible to do even better by employing the noncommutative Fourier techniques described in the remainder of the paper.

2. Fourier transform based on reducible representations of space groups

The main reason for using the group Fourier transform in this work is that the convolution theorem allows for the efficient computation of (2) (or equivalently, (3)). This provides an alternative technique for computing convolutions, and analyzing the limiting behavior of long walks by raising Fourier matrices to high powers rather than iterating convolution many times using (3) directly. For the theoretical foundations behind Fourier analysis on proper crystallographic space groups, see works such as [2,3,4,5,[11],[17],[20],22,29,37,38].

Since the walks of interest will always be finite, we can define a maximal length of interest, N, and then rather than doing calculations with the full lattice $\mathbb{L}$, we can do calculations with the quotient lattice $\mathbb{L}/N\mathbb{L}$. The group that acts on this lattice will be referred to here as the *lattice motion group*. This is the quotient group $G/N\mathbb{L}$. Functionally, the only difference between G and $G/N\mathbb{L}$ is that the latter is finite and translations larger than N steps in any spatial direction "wrap around," whereas G is infinite because translations continue on to infinity in all spatial directions. As long as the length of a walk satisfies $n < N/2$, then there is no difference in computing the distribution $f_n(g)$ with $g \in G$ or $g \in \mathfrak{g} \in G/N\mathbb{L}$ since there is no wrap-around effect. Here we will focus on the details of the Fourier transform for lattice motion groups and its application to polymer theory. Due to the simplicity of the form of reducible representations of these groups, we use them instead of irreducible representations. Although the classical

[†]Recall that the support of a function is the part of its domain corresponding to nonzero values of the function.

theory is built on irreducible representations, we show in the following sections that reducible representations can be used.

2.1. Basics of the representation theory of nite groups

A unitary representation of a finite group, such as $G/N\mathbb{L}$, is a mapping that takes elements of the group, g, and returns unitary matrices, $\pi(g, k)$, with the properties

$$\Pi(g_1 \circ g_2, k) = \Pi(g_1, k)\Pi(g_2, k) \quad \text{and} \quad \Pi(e, k) = \mathbb{I} \tag{4}$$

where e is the identity element of the group, k is an element of the dual space of the group, and $\mathbb{I}$ is the identity matrix with the same dimension as $\pi(g, k)$. Sometimes it will be convenient to write $\pi k(g)$ for $\pi(g, k)$ and sometimes we will leave off the argument and write π_k. The first equality in (4) is analogous to $e^{ik(\theta_1+\theta_2)} = e^{ik\theta_1}e^{ik\theta_2}$ which, together with unitarity and completeness, forms the foundations of all of the useful properties in Fourier analysis.

The matrix-valued functions $\pi k(g)$ in (4) are called *unitary representations*. Usually Fourier analysis on groups is defined based on *irreducible* unitary representations. These are matrices with the smallest possible dimensions. However, we use *reducible* representations that are easy to define. The dimension of these representations will be $|\mathbb{P}| \times |\mathbb{P}|$. Since $|\mathbb{P}|$ will not typically be a huge number (e.g., 4 for the square lattice, 12 for the diamond lattice, 24 for the cubic lattice), it is hardly worth the added effort to reduce them. However, since the classical theory only addresses irreducible representations, we need to verify that our Fourier transform for $G/N\mathbb{L}$ (which is defined in the following section) is in fact invertible.

Let $\{R_j\}$ for $j=1,\dots,|\mathbb{P}|$ denote the set of all rotations in the point group $\mathbb{P}$. Reducible representations of $G/N\mathbb{L}$ are matrices with entries of the form [7]

$$\Pi_{m,n}(g, \mathbf{k}) = e^{i(R_m\mathbf{k})\cdot(\mathbf{a})}\delta\left(R_n^T A^T R_m\right). \tag{5}$$

The δ function implies that $R_n^T A^T R_m = \mathbb{I}_\mathbb{P}$ (the identity matrix with the same dimensions as rotations in the proper point group $\mathbb{P}$), forcing the equality $A = R_m R_n^T$.

Let $g_1 = (A_1, a_1)$ and $g_2 = (A_2, a_2)$. The reducible representation defined in (5) evaluated at $g_1 \circ g_2$ is given by:

$$\Pi_{m,n}(g_1 \circ g_2, \mathbf{k}) = e^{i(R_m\mathbf{k})\cdot(A_1\mathbf{a}_2+\mathbf{a}_1)}\delta\left(R_n^T A_2^T A_1^T R_m\right). \tag{6}$$

Then let $A = \pi(g_1, \mathbf{k})$, $B = \pi(g_2, \mathbf{k})$, and $C = \pi(g_1, \mathbf{k})\pi(g_2, \mathbf{k})$. The entries of matrix C can be written as

$$C_{m,n} = \sum_j A_{m,j}B_{j,n} = \sum_j e^{i(R_m\mathbf{k})\cdot(\mathbf{a}_1)}\delta\left(R_j^T A_1^T R_m\right)e^{i(R_j\mathbf{k})\cdot(\mathbf{a}_2)}\delta\left(R_n^T A_2^T R_j\right). \tag{7}$$

Given the fact that $R_j = A_1^T R_m$, (7) can be rewritten as:

$$C_{m,n} = e^{i(R_m\mathbf{k})\cdot(\mathbf{a}_1)}e^{i(A_1^T R_m\mathbf{k})\cdot(\mathbf{a}_2)}\delta\left(R_n^T A_2^T A_1^T R_m\right), \tag{8}$$

or equivalently

$$C_{m,n}=e^{i(R_m\mathbf{k})\cdot(A_1\mathbf{a}_2+\mathbf{a}_1)}\delta\left(R_n^T A_2^T A_1^T R_m\right),$$

(9)

which indeed equals to the elements of (6).

2.2. Analysis: Fourier transform

The matrix elements of the Fourier transform of a function on a lattice motion group can be defined as:

$$\widehat{f}(\mathbf{k})_{m,n}=\sum_{g\in G/N\mathbb{L}} f(g)\,\Pi\left(g^{-1},\mathbf{k}\right)=\sum_{A\in\mathbb{P}}\sum_{\mathbf{a}\in\mathbb{L}/N\mathbb{L}} f(\mathbf{a},A)\,e^{i(R_m\mathbf{k})\cdot(-A^T\mathbf{a})}\delta\left(R_n^T A R_m\right).$$

(10)

Using the properties of the delta function, (10) can be rewritten as

$$\widehat{f}(\mathbf{k})_{m,n}=\sum_{\mathbf{a}\in\mathbb{L}/N\mathbb{L}} f\left(\mathbf{a},R_n R_m^T\right)e^{-i\left(\mathbf{k}^T R_m^T R_m R_n^T\mathbf{a}\right)}$$

(11)

where $\mathbf{k}$ is drawn from the reciprocal lattice (see the appendix). By setting $\mathbf{u}=R_n^T\mathbf{a}$, (11) becomes

$$\widehat{f}(\mathbf{k})_{m,n}=\sum_{\mathbf{u}\in\mathbb{L}/N\mathbb{L}} f\left(R_n\mathbf{u},R_n R_m^T\right)e^{-i(\mathbf{k}\cdot\mathbf{u})}.$$

(12)

Equation (12) can be identified as a d-dimensional space DFT-transform, where d is the dimension of the lattice. That is, $|\mathbb{L}/N\mathbb{L}|=N^d$. The sum over $\mathbb{L}/N\mathbb{L}$ is a nested sum of d one-dimensional sums from $j=0$ to $N-1$. The matrix $\widehat{f}(\mathbf{k})$ is a $|\mathbb{P}|\times|\mathbb{P}|$ matrix where $|\mathbb{P}|$ is the number of rotations in the proper point group $\mathbb{P}$. We can reduce the computational cost by using the classical fast Fourier transform (FFT) to compute (12). The FFT is standard in many numerical/computational software packages.

In (12), the sum $\sum_{\mathbf{u}\in\mathbb{L}/N\mathbb{L}} f\left(R_n\mathbf{u},R_n R_m^T\right)e^{-i(\mathbf{k}\cdot\mathbf{u})}$ is calculated over $\mathbf{u}$. The FFT is calculated with respect to $\mathbf{u}$ and the non-zero terms of the summation occur on the support of $f\left(R_n\mathbf{u},R_n R_m^T\right)$. Since the term $R_n\mathbf{u}$ is actually the $\mathbf{a}$ term, we have a function of $\mathbf{a}$ calculated over the $\mathbf{u}$, where $\mathbf{u}=R_n^T\mathbf{a}$. If we use the $f(\mathbf{a},A)$ as an input argument in the FFT function in standard packages, then since f is a function of $\mathbf{a}$ and the FFT should be computed over $\mathbf{u}$, the FFT must be called as $FFTD\left(f\left(R_n^T\mathbf{a},R_n R_m^T\right)\right)$, where the term $FFTD$ is the pseudocode of the d-dimensional direct FFT function. Our goal is to use this Fourier transform to compute convolutions and to reconstruct $f(\mathbf{a},A)$ from (12).

2.3. Convolutions using the Fourier transform

Our whole reason for wanting to compute Fourier transforms on space groups is that the property (4) immediately implies that

$$\widehat{f_1 * f_2}(\mathbf{k})=\widehat{f_2}(\mathbf{k})\,\widehat{f_1}(\mathbf{k}).$$

(13)

This can be proven abstractly without going into the detailed structure of the representations in (5), as long as (4) holds. We know (4) to be true from the analysis provided in the equations (7)-(9).

By definition we have

$$\left(\widehat{f_1 * f_2}\right)(\Pi_{\mathbf{k}}) = \sum_{g \in G'}\left(\sum_{h \in G'} f_1(h) f_2\left(h^{-1} \circ g\right)\right)\Pi_{\mathbf{k}}\left(g^{-1}\right)$$

where $G'=G/N\mathbb{L}$. By setting $p = h^{-1} \circ g$ and substituting $g = h \circ p$, we have the following expression

$$\sum_{p \in G'}\sum_{h \in G'} f_1(h) f_2(p) \Pi_{\mathbf{k}}\left(p^{-1} \circ h^{-1}\right).$$

By taking into account the homomorphism property of $\pi_{\mathbf{k}}$ and the commutativity of scalar-matrix multiplication and summation, this can be split into

$$\left(\sum_{p \in G'} f_2(p) \Pi_{\mathbf{k}}\left(p^{-1}\right)\right)\left(\sum_{h \in G'} f_1(h) \Pi_{\mathbf{k}}\left(h^{-1}\right)\right) = \widehat{f_2}(\Pi_{\mathbf{k}}) \widehat{f_1}(\Pi_{\mathbf{k}}).$$

This works regardless of whether or not the representations are irreducible.

Of course, such a convolution theorem has value only when we can recover a function from its Fourier transform. This is verified in the following subsection.

2.4. Synthesis: Inverse Fourier transform

The function $f(\mathbf{a}, A)$ can be recovered from the set of Fourier transform matrices $\{\widehat{f}(\mathbf{k})\}$ by the following formula:

$$f(\mathbf{a}, A) = \frac{1}{|\mathbb{P}|}\sum_{m,n=0}^{|\mathbb{P}|-1}\frac{1}{N^d}\sum_{\mathbf{k}}\widehat{f}(\mathbf{k})_{m,n}e^{i\mathbf{k}\cdot R_n^T\mathbf{a}}\delta\left(R_n^T A R_m\right).$$

(14)

It may not be obvious a priori that a function $f(\mathbf{a}, A)$ on a lattice motion group can be recovered from the Fourier transform in (10), since the representations are reducible. We can prove that (14) does recover $f(\mathbf{a}, A)$ by first writing:

$$g(\mathbf{a}, A) = \frac{1}{|\mathbb{P}|}\sum_{m,n=0}^{|\mathbb{P}|-1}\frac{1}{N^d}\sum_{\mathbf{k}}\sum_{\mathbf{u}\in\mathbb{L}/N\mathbb{L}} f\left(R_n\mathbf{u}, R_n R_m^T\right)e^{-i(\mathbf{k}\cdot\mathbf{u})}e^{i\mathbf{k}\cdot R_n^T\mathbf{a}}\delta\left(R_n^T A R_m\right).$$

(15)

We will see if $g(\mathbf{a}, A) = f(\mathbf{a}, A)$. By setting $v = R_n\mathbf{u} \Rightarrow \mathbf{u} = R_n^T v$, (15) becomes

$$g\left(\mathbf{a},A\right)=\frac{1}{|\mathbb{P}|}\sum_{m,n=0}^{|\mathbb{P}|-1}\frac{1}{N^d}\sum_{\mathbf{k}}\sum_{v\in\mathbb{L}/N\mathbb{L}}f\left(v,R_nR_m^T\right)e^{-iR_n\mathbf{k}\cdot(v-a)}\delta\left(R_n^TAR_m\right).$$

(16)

By setting $\mathbf{k}' = R_n\mathbf{k}$, this becomes

$$g\left(\mathbf{a},A\right)=\frac{1}{|\mathbb{P}|}\sum_{m,n=0}^{|\mathbb{P}|-1}\frac{1}{N^d}\sum_{\mathbf{k}'}\sum_{v\in\mathbb{L}/N\mathbb{L}}f\left(v,R_nR_m^T\right)e^{-i\mathbf{k}'\cdot(v-a)}\delta\left(R_n^TAR_m\right),$$

(17)

which obviously takes the form

$$g\left(\mathbf{a},A\right)=\frac{1}{|\mathbb{P}|}\sum_{m,n=0}^{|\mathbb{P}|-1}f\left(\mathbf{a},R_nR_m^T\right)\delta\left(R_n^TAR_m\right)=f\left(\mathbf{a},A\right).$$

(18)

It is convenient to calculate the inverse Fourier transform for a lattice motion group using the classical Abelian inverse FFT transform in d-dimensional lattices for the translational part, and computing the rotational part by brute-force. That is, since (14) can be rewritten as

$$f\left(\mathbf{a},A\right)=\frac{1}{|\mathbb{P}|}\sum_{m,n=0}^{|\mathbb{P}|-1}\frac{1}{N^d}\sum_{\mathbf{k}}\widehat{f}(\mathbf{k})_{m,n}e^{iR_n\mathbf{k}\cdot\mathbf{a}}\delta\left(R_n^TAR_m\right)$$

(19)

and since we can set $\mathbf{k}_1=R_n\mathbf{k} \Rightarrow \mathbf{k}=R_n^T\mathbf{k}_1$, (19) becomes

$$f\left(\mathbf{a},A\right)=\frac{1}{|\mathbb{P}|}\sum_{m,n=0}^{|\mathbb{P}|-1}\frac{1}{N^d}\sum_{\mathbf{k}_1}\widehat{f}(R_n\mathbf{k}_1)_{m,n}e^{i\mathbf{k}_1\cdot\mathbf{a}}\delta\left(R_n^TAR_m\right).$$

(20)

The term $\frac{1}{N^d}\sum_{\mathbf{k}_1}\widehat{f}(R_n\mathbf{k}_1)_{m,n}e^{i\mathbf{k}_1\cdot\mathbf{a}}$ can be easily calculated by using software packages containing the inverse Fourier transform. Again we note, as in the case of the Fourier transform, that since the sum is over $\mathbf{k}_1$, and $\widehat{f}$ is a function of $\mathbf{k}$, the pseudocode would be of the form $IFFTD\left(\widehat{f}(R_n\mathbf{k})\right)_{m,n}$. Since we have the term $\delta\left(R_n^TAR_m\right) \Rightarrow R_m=A^TR_n$, we can simply calculate $f(\mathbf{a}, A)$ using $\frac{1}{|\mathbb{P}|}\sum_n IFFTD\left(\widehat{f}(R_n\mathbf{k})_{\mathrm{index}\left(A^T,R_n\right),n}\right)$, where "*index*" calls the group table for the proper point group of rotational b symmetries of the lattice, $\mathbb{P}$.

2.5. Computational cost

The new feature of the methods presented in this paper and that of [40] is that the methods can efficiently generate exact statistics (to within machine precision) corresponding to *all* of the exponential number of conformations. This becomes more and more convenient when the number of segments increases. While previous methods relied on sampling, the new method presented here finds statistics for all the possible conformations in a very short time. A natural question to ask is if, and how much, the Fourier transform speeds up the computation of convolutions on proper space groups. The Fourier transform based on irreducible unitary

representations will naturally be the fastest, but it also requires the most pre-processing in terms of determining all of the IURs.

Consider the case of a homogeneous polymer in which the single-step distribution function describing allowable conformational motions going from residue i to $i + 1$ is $f_1(g)$. The end-to-end distribution in positions and orientations for a polymer of length L residues will then be the n-fold convolution $f_L(g) = (f_1 * f_1 * \ldots * f_1)(g)$. This chain will fit within periodized lattice with $N > 2L$ without wrapping the polymer around to the other side.

For each fixed value of $m, n \in \mathbb{P}$, the convolution over the periodized translational lattice can be computed via lattice translational FFT in $O\left(N^d \log_2 N^d\right)$ operations where $|\mathbb{L}/N\mathbb{L}| = N^d$. This means that the cost for computing over all values of m and n will be $|\mathbb{P}|^2$ times this value, or $O\left(|\mathbb{P}|^2 \cdot N^d \log_2 N^d\right)$. This means that it is asymptotically more costly to perform one convolution by FFT than a single one-step convolution. However, if the goal is to find only $f_{L_{max}}(g)$ without all of the intermediate functions f_i for $i = 1, \ldots, L_{max} - 1$, then the method is faster. The Fourier matrices $\widehat{f_{L_{max}}}$ can be computed by using the Fourier matrix $\widehat{f_1}$. We can use the fact that $\widehat{f_2} = \left(\widehat{f_1}\right)^2$, $\widehat{f_4} = \left(\widehat{f_2}\right)^2$ and so forth. Hence we do not have to compute $(L_{max} - 1)$-times matrix multiplications to have $\widehat{f_{L_{max}}}$. Roughly $(\log_2 L_{max})$-times matrix multiplications will be enough. Consequently, the computation of $\widehat{f_{L_{max}}}$ requires $O\left(|\mathbb{P}|^3 L_{max}^d \log_2 L_{max}^d\right) = O\left(|\mathbb{P}|^3 N^d \log_2 N^d\right)$ operations, N is determined to be proportional to L_{max}. Since $|\mathbb{P}|$ is independent of the number of segments, the dominant complexity in overall computation is $O\left(N^d \log_2 N^d\right)$.

3. Numerical results

In this section we demonstrate our method by calculating the probability distribution functions for non-reversal random walks of up to 400 segments on the square lattice and up to 20 segments on the cubic and diamond lattices.

First, we develop the array which defines the supports of the probability distribution functions on the square, cubic and diamond lattices. Taking into account the final number of segments that we want to reach (400, 20 and 20), we define arrays of dimensions (801, 801, 4) for the square, (41, 41, 41, 24) for the cubic and (41, 41, 41, 12) for the diamond lattice. The first two indices (for the square) and the first three (for the cubic and diamond) denote the Cartesian coordinates on the lattice whereas the last array coordinate denotes the index of the element of the point group describing discrete rotations. In order to avoid aliasing, the arrays for the Cartesian coordinates must be at least $(2N + 1, 2N + 1)$ for the square lattice and $(2N + 1, 2N + 1, 2N + 1)$ for the cubic and diamond lattices where N is the number of segments.

We use the periodized lattice. This can be obtained using the $mod_{2N + 1}$ operation. For example, in the square lattice, the coordinate of $(-1, 0)$ for the periodized lattice would be $(mod(-1, 801), mod(0, 801)) = (800, 0)$. After we obtain the support arrays of the functions in the periodized lattice, we compute the fast Fourier transform thereof using the reducible representations we described above. The array for the Fourier transform in the square lattice will be in dimensions of (401, 401, 4, 4), and in the cubic lattice will be (41, 41, 41, 24, 24) and in the diamond lattice (41, 41, 41, 12, 12). One issue with the Fourier transform is the Gibbs peaks that occur for large numbers of segments (greater than 30). Classically these peaks occur when computing the Fourier transform of discontinuous functions. The probability distribution functions of random walks can exhibit sharp changes in values from one point in the lattice motion group

to another. We deal with the problem by normalizing all the values by the biggest value. We round off all the values less than 10^{-15} to zero. After finding the Fourier transform, $\widehat{f_1}$, we compute the Fourier transform for the longer segments. We compute $\widehat{f_n}(\Pi_{\mathbf{k}}) = \left(\widehat{f_1}(\Pi_{\mathbf{k}})\right)^n$ for all possible $\mathbf{k}$, n is the number of segments. Finally, inverse Fourier transform gives $f_n(g)$.

3.1. Square lattice

In Figure 1, we present the distribution of the end point of 20-segment NRRW on the square lattice. The probability density is depicted by the size and the darkness of the circles. All the 3^{20} possible end points are obtained and from there we can extract various statistics including the probability distribution of the end-to-end distance. Although we limited the number of segments as 20 in this example, we can increase this to a large value as long as the computer memory allows. In the following test, we use the 400-segment NRRWs.

Several important statistics, that are calculated in the literature [13,14,12,23,26,33], and give information about the shape (width, skewness) of the distribution functions for R^2, are $\frac{\langle R^4 \rangle}{\langle R^2 \rangle^2}$, $\frac{\langle R^6 \rangle}{\langle R^2 \rangle^3}$ and $\frac{\langle R^8 \rangle}{\langle R^2 \rangle^4}$, where R is the end-to-end distance of the random walk. We compute these quantities using the end points obtained by our method, and we compare them with the results by the method described in [32]. When we compute the quantities, the NRRWs with from 1 to 400 segments are considered. For convenience, we will refer to $\frac{\langle R^4 \rangle}{\langle R^2 \rangle^2}$ as RATIO4, $\frac{\langle R^6 \rangle}{\langle R^2 \rangle^3}$ as RATIO6 and $\frac{\langle R^8 \rangle}{\langle R^2 \rangle^4}$ as RATIO8. In Figures 2(a),3(a), and 4(a), we respectively compare RATIO4, RATIO6 and RATIO8 obtained using our method (crosses) and by the method described in [32] (a solid line). In addition, Figures 2(b),3(b), and 4(b) respectively show the difference at every number of segments. This difference is defined as $(b - a)/a$, where a and b are the statistics (RATIO4, RATIO6 or RATIO8) obtained by the Mattice method [32] and our method, respectively.

Another statistical quantity of great importance is that of the ring closure. The approximate probability of the ring closure can be calculated analytically by the method of Jacobson and Stockmayer [21]. The term "ring closure" simply means the probability for a random walk to terminate in the origin. For the square lattice, in order for a walk to terminate in the origin, it has to have even number of segments and the number of segments has to be greater or equal than four. For the case of the square lattice, which is a two-dimensional lattice, the Jacobson-Stockmayer result reduces to the formula $\frac{1}{\pi n}$ [21]. We count the number of walks that are obtained by our Fourier method and terminate in the origin. Then we divide that number by the total number of random walks to have the probability of the ring closure. This probability is compared with the Jacobson-Stockmayer result, $\frac{1}{\pi n}$.

Figure 5(a) shows the probabilities of ring closure on the square lattice. The two results are obtained by the Jacobson-Stockmayer formula (a solid line) and our method (crosses). Figure 5(b) shows the difference between the two results in Figure 5(a). The difference is defined as $(b-a)/a$, where a and b are the probability of ring closure obtained by the Jacobson-Stockmayer method [21] and our method, respectively. The Jacobson-Stockmayer formula provides the probability of ring closure that is more accurate with larger n. The fact that the difference in

Figure 5(b) is getting closer to zero means that the probability of ring closure obtained by our method is reliable.

It is important to note that the main purpose of this comparison between the various statistics is to verify that our method accurately generates the NRRWs. Using the NRRWs obtained by our method, any kinds of statistics of the random walks can be computed, while the methods such as the Mattice method and the Jacobson-Stockmayer method can compute only the specific statistics.

3.2. Cubic lattice

As in the square lattice, we can apply our method to the NRRWs on the cubic lattice. Figure 6 shows the distribution of the end points of 10-segment NRRWs on the cubic lattice. Figures 7(a), 8(a), and 9(a) show the statistics of the end-to-end distance obtained using our method (crosses) and by the method described in [32] (a solid line) for up to 20 segments. In addition, Figures 7(b), 8(b), and 9(b) show the corresponding difference at every number of segments. This difference is defined in the same way as in the square lattice.

The probability of the ring closure that was calculated analytically by Jacobson and Stockmayer [21] is $\left(\frac{3}{2\pi n}\right)^{3/2}$ for the cubic lattice. In order for a walk on the cubic lattice to terminate in the origin, it has to have even number of segments and the number of segments has to be greater than or equal to four. Figure 10(a) shows the probabilities of the ring closure on the cubic lattice. The two results are obtained by the Jacobson-Stockmayer method (a solid line) and our method (crosses). Figure 10(b) shows the difference between the two results in Figure 10(a). We define this difference in the same way as in the square lattice.

3.3. Diamond lattice

As in the above two lattices, we can apply our method to the NRRWs on the diamond lattice. One distinctive feature of the diamond lattice is that this lattice is 2-definable [35]. Therefore, we start with $\widehat{f_2}$ as a smallest step and compute the NRRWs of the even number of segments. Figure 11 shows the distribution of the end points of 10-segment NRRWs on the diamond lattice. Figures 12(a),13(a), and 14(a) show the statistics of the end-to-end distance obtained using our method (crosses) and by the method described in [32] (a solid line) for up to 20 segments. In addition, Figures 12(b), 13(b), and 14(b) show the corresponding difference at every number of segments. This difference is defined in the same way as in the above two lattices.

The probability of the ring closure calculated by Jacobson and Stockmayer [21] is $\left(\frac{3}{2\pi n}\right)^{3/2}$ for the diamond lattice. In order for a walk on the diamond lattice to terminate in the origin, it has to have even number of segments and the number of segments has to be greater than or equal to six. Figure 15(a) shows the probabilities of the ring closure on the diamond lattice. The two results are obtained by the Jacobson-Stockmayer method (a solid line) and our method (crosses). Figure 15(b) shows the difference between the two results in Figure 15(a). We define this difference in the same way as in the above two lattices.

4. Conclusions

Lattice random walks with no immediate reversals have applications in several various fields such as polymer science, finance, crystallography, and robotics. In particular, we are interested in studying properties of polymers that depend mainly on their conformations. The statistics of these conformations are studied by simulating them as lattice random walks. As the number

of lattice segments increases, the number of all possible conformations increases exponentially. In this paper we developed a method for calculating the distribution of the distal ends of the N-segment NRRWs in $O\left(N^2 \log_2 N\right)$ computational cost for the square lattice and $O\left(N^3 \log_2 N\right)$ computational costs for the spatial lattices (e.g., cubic and diamond). This is a significant improvement compared to the exponential computational cost required by brute-force enumeration, or the random sampling that is often used (e.g., Monte Carlo methods). Our method generated the distribution of the distal end of long NRRWs using convolution of short NRRWs. This convolution was implemented using the group Fourier transform for the lattice motion group.

In order to verify our method, we used our method to study statistical properties of lattice random walks, and then compared them with the statistical quantities that other existing methods computed. We considered the statistics of the end-to-end distance, which are $\dfrac{\left\langle R^4 \right\rangle}{\left\langle R^2 \right\rangle^2}$, $\dfrac{\left\langle R^6 \right\rangle}{\left\langle R^2 \right\rangle^3}$ and $\dfrac{\left\langle R^8 \right\rangle}{\left\langle R^2 \right\rangle^4}$, where R is the end-to-end of distance the random walks. First, these quantities were computed using the position information of the distal end of the random walks that are obtained using our method. Then the same statistical quantities were computed using the Mattice method [32]. We also considered the probability of the ring closure. This quantity was computed using two methods: Jacobson-Stockmayer theory [21] and our method. All the comparison between the statistical quantities that we considered here verified that our method generates the accurate distribution of the conformations of the non-reversible random walks.

Acknowledgments

Appendix: Reciprocal lattices for square, cubic and diamond lattices

A lattice is an infinite array of regularly spaced points in $\mathbb{R}^d$. Given a linearly independent set of not-necessarily-orthogonal basis vectors in $\mathbb{R}^d$, i.e.,

$$\{\mathbf{b}_1, \dots, \mathbf{b}_d\} \quad \text{with} \quad \det\left[\mathbf{b}_1, \dots, \mathbf{b}_d\right] \neq 0,$$

a lattice is defined as

$$\mathbb{L} = \left\{ \sum_{i=1}^{d} m_i \mathbf{b}_i \mid m_i \in \mathbb{Z} \right\}$$

where $\mathbb{Z}$ denotes the integers. The vectors $\mathbf{k}$, for the case of the multidimensional lattice Discrete Fourier Transform are defined to be of the form

$$\mathbf{k} = \sum_{j=1}^{d} k_j \widehat{\mathbf{b}}_j$$

(21)

where the set of vectors $\left\{\widehat{\mathbf{b}}_j\right\}$ are defined by the conditions

$$\mathbf{b}_i \cdot \widehat{\mathbf{b}}_j = 2\pi\delta_{ij}. \tag{22}$$

Written for all $i, j \in \{1,\ldots, d\}$ this becomes

$$\begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \vdots \\ \mathbf{b}_d^T \end{bmatrix} \left[\widehat{\mathbf{b}}_1, \widehat{\mathbf{b}}_2, \ldots, \widehat{\mathbf{b}}_d \right] = 2\pi\mathbb{I}.$$

From this, it is clear that a set of vectors $\left\{ \widehat{\mathbf{b}}_j \right\}$ can always be found when a basis $\{\mathbf{b}_i\}$ is given, since

$$\left[\widehat{\mathbf{b}}_1, \widehat{\mathbf{b}}_2, \ldots, \widehat{\mathbf{b}}_d \right] = 2\pi [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d]^{-T}$$

is well defined.

The set of vectors $\left\{ \widehat{\mathbf{b}}_j \right\}$ forms a basis for a new lattice, called the *reciprocal lattice*, denoted as

$$\widehat{\mathbb{L}} = \left\{ \sum_{i=1}^{d} m_i \widehat{\mathbf{b}}_i \mid m_i \in \mathbb{Z} \right\}.$$

It is possible for a lattice to be self-reciprocal, i.e., $\widehat{\mathbb{L}} = \mathbb{L}$, as in our cases of the unit square, the cubic and the diamond lattices.

References

[1]. Alexandrowicz Z. Monte carlo of chains with excluded volume - a way to evade sample attrition. Journal of Chemical Physics 1969;51:561–565.

[2]. Blokker E. A theory for the construction of the irreducible representations of finite groups. International Journal of Quantum Chemistry 1972;6(5):925–933.

[3]. Blokker E. A theory for the construction of the irreducible representations of finite groups. II. International Journal of Quantum Chemistry 1973;7(6):1091–1097.

[4]. Bradley, CJ.; Cracknell, AP. Mathematical Theory of Symmetry in Solids. Oxford U. Press; 1971.

[5]. Chechin GM, Popov VP, Raspopov VN. The irreducible representations of hexagonal space-groups. Kristallogra ya 1980;25(4):661–674.

[6]. Chirikjian, GS. Stochastic Models, Information Theory, and Lie Groups. Vol. Vols. I and II. Birkhäuser; Boston: 2009/11.

[7]. Chirikjian, GS.; Kyatkin, AB. Engineering Applications of Noncommutative Harmonic Analysis. CRC Press; Boca Raton, FL: 2001.

[8]. Diaconis P, Rockmore D. Efficient computation of the Fourier transform on finite groups. Journal of the American Mathematical Society 1990;3(2):297–332.

[9]. Diaconis P, Saloff-Coste L. Comparison techniques for random walk on finite groups. The Annals of Probability 1993;21(4):2131–2156.

[10]. Erschler A. On drift and entropy growth for random walks on groups. The Annals of Probability 2003;31(3):1193–1204.

[11]. Evarestov, RA.; Smirnov, VP. Site Symmetry in Crystals: Theory and Applications. 2nd edition. Springer-Verlag; Berlin: 1997.

[12]. Flory PJ. Foundations of rotational isomeric state theory and general methods for generating configurational averages. Rotational Isomeric State Theory and Methods 1974;7(3):381–392.

[13]. Flory, PJ. Statistical Mechanics of Chain Molecules. Hanser/Gardner Publications; Cincinnati, OH: 1985.

[14]. Flory PJ, Jernigan RL. Second and fourth moments of chain molecules. Journal Of Chemical Physics 1965;42(10):3509–3519.

[15]. Furstenberg H. Random walks and discrete subgroups of Lie groups. Advances in Probability and Related Topics 1971;1:1–63.

[16]. Grenander, U. Probabilities on Algebraic Structures. Dover: 2008.

[17]. Hahn, T. International tables for crystallography. Brief teaching edition of volume A, Space-group symmetry. Kluwer Academic Publishers; Boston, MA: 1996.

[18]. He SQ, Scheraga HA. Macromolecular conformational dynamics in torsional angle space. Journal of Chemical Physics 1998;108:271–286.

[19]. Heyer, H. Probability Measures on Locally Compact Groups. Springer-Verlag; 1977.

[20]. Iversen, B. Lectures on Crystallographic Groups. Matematisk Institut, Aarhus Universitet; 1990.

[21]. Jacobson H, Stockmayer WH. Intramolecular reaction polycondensations. I. the theory of linear systems. Journal of Chemical Physics 1950;18(12):1600–1606.

[22]. Janssen, T. Crystallographic Groups. North Holland/American Elsevier; New York: 1972.

[23]. Jernigan RL, Flory PJ. Moments of chain vectors for models of polymer chains. Journal of Chemical Physics 1969;50(10):4178–4185.

[24]. Kesten H. Symmetric random walks on groups. Transactions of the American Mathematical Society 1959;92(2):336–354.

[25]. Kesten H. On the number of self-avoiding walks. Journal of Mathematical Physics 1963;4(7):960–969.

[26]. Kotelyanskii, M.; Theodorou, DN. Simulation methods for Polymers. Marcel Dekker Inc.; New York, NY: 2004.

[27]. Kuhn W. Concerning the shape of thread shapes molecules in solution. Kolloid-Zeitschrift 1934;68:2–15.

[28]. Lawler, GF. Intersections of Random Walks. Birkhäuser; Boston: 1996.

[29]. Mackey, GW. The Theory of Unitary Group Representations. The University of Chicago Press; Chicago: 1976.

[30]. Madras, N.; Slade, G. The self-avoiding walk. Birkhäuser; Boston: 1996.

[31]. Mark JE, Abou-Hussein R, Sen TZ, Kloczkowski A. Some simulations on filler reinforcement in elastomers. Polymer 2005;46:8894–8904.

[32]. 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):1956–1959.

[33]. Mattice, WL.; Suter, UW. Conformational Theory of Large Molecules, the Rotational Isomeric State Model in Macromolecular Systems. Wiley; New York, NY: 1994.

[34]. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of state calculations by fast computing machines. Journal of Chemical Physics 1953;21:1087–1092.

[35]. Meyer, P. Lattice geometries. http://www.hermetic.ch/compsci/lattgeom.htm

[36]. Montroll EW, Weiss GH. Random walks on lattices. II. Journal of Mathematical Physics 1965;6(2):167–181.

[37]. Neto N. Irreducible representations of space groups. Acta Crystallographica 1973;29:464–472.

[38]. Neto N. Numerical-calculations of irreducible representations of space groups. Computer Physics Communications 1975;9(4):231–246.

[39]. Sharaf MA, Mark JE. Monte carlo simulations on the effects of nanoparticles on chain deformations and reinforcement in amorhous polyethylene networks. Polymer 2004;45:3943–3952.

[40]. Skliros A, Chirikjian GS. Torsional random walk statistics on lattices using convolution on crystallographic motion groups. Polymer 2007;48(7):2155–2173. [PubMed: 17898862]

[41]. Spitzer, F. Principles of Random Walk. 2nd edition. Springer; 1976.

[42]. Vanderzande, C. Lattice models of polymers. Cambridge University Press; 1998.

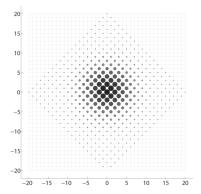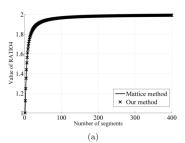[43]. Weiss, GH. Aspects and applications of the random walk. North Holland: 1994.

Figure 1.
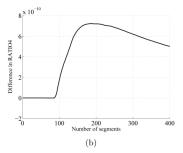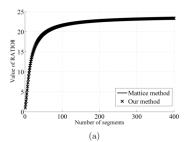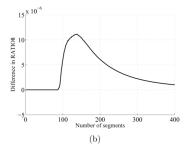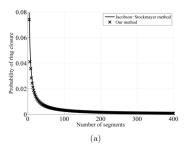Distribution of the end points of 20-segment NRRWs on the square lattice

Figure 2.
RATIO4 = $\langle R^4 \rangle / \langle R^2 \rangle^2$ in the square lattice. (a) Comparison of the results obtained by Mattice method (a solid line) and our method (crosses). (b) Difference between the two results in (a).

Figure 3.
RATIO6 $= \langle R^6 \rangle / \langle R^2 \rangle^3$ in the square lattice. (a) Comparison of the results obtained by Mattice method (a solid line) and our method (crosses). (b) Difference between the two results in (a).

Figure 4.
RATIO8 = $\langle R^8 \rangle / \langle R^2 \rangle^4$ in the square lattice. (a) Comparison of the results obtained by Mattice method (a solid line) and our method (crosses). (b) Difference between the two results in (a).
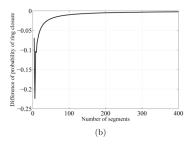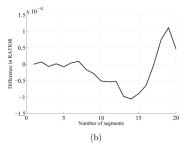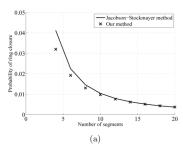
Figure 5.
Probability of ring closure on the square lattice (a) Comparison of the results obtained by Jacobson-Stockmayer method (a solid line) and our method (crosses). (b) Difference between the two results in (a).
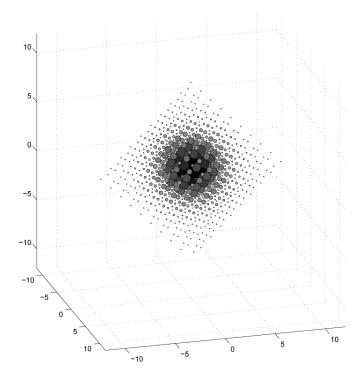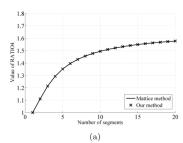
Figure 6.
Distribution of the end points of 10-segment NRRWs on the cubic lattice
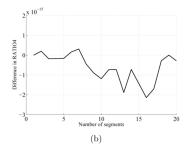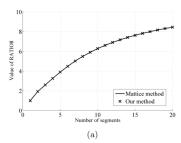
Figure 7.
RATIO4 = $\langle R^4 \rangle / \langle R^2 \rangle^2$ in the cubic lattice. (a) Comparison of the results obtained by Mattice method (a solid line) and our method (crosses). (b) Difference between the two results in (a).
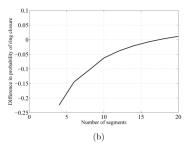
Figure 8.
RATIO6 = $\langle R^6 \rangle / \langle R^2 \rangle^3$ in the cubic lattice. (a) Comparison of the results obtained by Mattice method (a solid line) and our method (crosses). (b) Difference between the two results in (a).
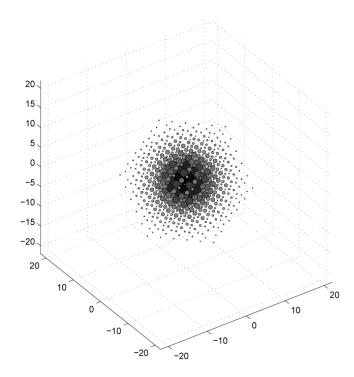
Figure 9.
RATIO8 $= \langle R^8 \rangle / \langle R^2 \rangle^4$ in the cubic lattice. (a) Comparison of the results obtained by Mattice method (a solid line) and our method (crosses). (b) Difference between the two results in (a).
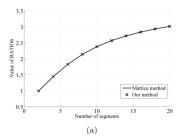
Figure 10.
Probability of ring closure on the cubic lattice (a) Comparison of the results obtained by Jacobson-Stockmayer method (a solid line) and our method (crosses). (b) Difference between the two results in (a).

Figure 11.
Distribution of the end points of 10-segment NRRWs on the diamond lattice

Figure 12.
RATIO4 = $\langle R^4 \rangle / \langle R^2 \rangle^2$ in the diamond lattice. (a) Comparison of the results obtained by Mattice method (a solid line) and our method (crosses). (b) Difference between the two results in (a).
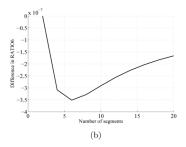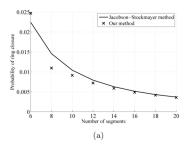
Figure 13.
RATIO6 $= \langle R^6 \rangle / \langle R^2 \rangle^3$ in the diamond lattice. (a) Comparison of the results obtained by Mattice method (a solid line) and our method (crosses). (b) Difference between the two results in (a).

Figure 14.
RATIO8 $= \langle R^8 \rangle / \langle R^2 \rangle^4$ in the diamond lattice. (a) Comparison of the results obtained by Mattice method (a solid line) and our method (crosses). (b) Difference between the two results in (a).
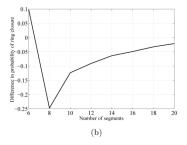
Figure 15.
Probability of ring closure on the diamond lattice (a) Comparison of the results obtained by
Jacobson-Stockmayer method (a solid line) and our method (crosses). (b) Difference between
the two results in (a).