Parallel algorithm for calculation of the exact partition function of a lattice polymer

Jae Hwan Lee, Seung-Yeon Kim, Julian Lee

A School of Systems Biomedical Science and Department of Bioinformatics and Life Science, Soongsil University, Seoul 156-743, Republic of Korea
B School of Liberal Arts and Sciences, Chungju National University, Chungju 380-702, Republic of Korea
C Department of Pharmaceutical Chemistry and Graduate Group in Biophysics, University of California, San Francisco, CA 94158, USA

Abstract

We develop a parallel algorithm that calculates the exact partition function of a lattice polymer, by enumerating the number of conformations for each energy level. An efficient parallelization of the calculation is achieved by classifying the conformations according to the shape of the box spanned by a conformation, and enumerating only those in a given box at a time. The calculation time for each box is reduced by preventing the conformations related by symmetries from being generated more than once. The algorithm is applied to study the collapse transition of a lattice homopolymer on a square lattice, by calculating the specific heat for chain lengths up to 36.

Keywords:
Lattice polymer
Exact enumeration
Parallel computation

1. Introduction

Polymers on discrete lattices serve as a simple toy model of a polymer [1–6]. By introducing hydrophobic inter-monomer interaction, the lattice model can be used to study the collapse transition of a polymer [7–26]. Various quantities such as radius of gyration, end-to-end distance, and specific heat have been calculated, both using Monte Carlo samplings [8,11,16–22,24–26] and exact enumeration [9,12,14,23]. Although the length of chain that can be studied using exact enumeration is much less than that of the Monte Carlo sampling methods, the exactness of the calculation enables one to use powerful extrapolation methods to study the behavior of the lattice polymers in the limit of infinite chain length.

In the current study, we develop a method for calculating the exact partition function of a lattice polymer. The partition function is the most basic quantity from which all the important thermodynamic properties can be calculated. The number of states for each energy $E$, $\Omega(E)$, contains all the information needed for the calculation of the partition function $Z$, which is given by

$$Z(\beta) = \sum_E \Omega(E) e^{-\beta E},$$

where $\beta = 1/k_B T$ with the Boltzmann constant $k_B$ and temperature $T$. In particular, it can be used for studying the collapse phase transition of the lattice polymer by observing behaviors of various quantities such as the specific heat,

$$C(T, N) = \frac{\partial E}{\partial T} = \beta^2 \frac{\partial^2 \ln Z}{\partial \beta^2},$$

with increasing $N$.

Since the number of conformations is finite in the case of a lattice polymer, the partition function can be made a polynomial in $e^{-\beta}$ by considering models with integer values of $E$. To elaborate, we consider a polymer on regular lattice such as two-dimensional square or three-dimensional cubic lattices. Then the Hamiltonian for a heteropolymer is given by

$$\mathcal{H} = - \sum_{i<j} \epsilon(a_i, a_j) \Delta(r_i, r_j),$$

where

$$\Delta(r_i, r_j) = \begin{cases} 1 & (|i-j| > 1 \text{ and } |r_i - r_j| = 1) \\ 0 & \text{(otherwise)}, \end{cases}$$

$a_i$ is the type of the $i$-th monomer, and $\epsilon(a, b)$ is the interaction energy between the monomers of type $a$ and $b$. By taking $\epsilon(a, b)$ to be integer multiples of a unit energy $\epsilon$, $\epsilon(a, b) = n(a, b)\epsilon$, the partition function is expressed as a polynomial:

$$Z = \sum_{\{K(a,b)\}} \tilde{\Omega}(\{K(a,b)\}) z^{\sum_n n(a,b)K(a,b)} = \sum_K \Omega(K) z^K,$$

where $z \equiv \exp(\beta \epsilon)$, $\tilde{\Omega}(\{K(a,b)\})$ is the number of polymer conformations with contact numbers $\{K(a,b)\}$,
\[ \Omega(K) = \sum \hat{\Omega}\left(\{x(a, b)\}\right) \]

is the number of conformations with energy \( E = -K\epsilon \). The homopolymer composed of only one kind of monomers is a special case where \( n(a, b) = 1 \) regardless of \( a, b \), with \( K \) now being the number of intra-chain contacts.

In this work, we develop an efficient parallel algorithm for calculating the exact partition function of a lattice polymer by enumerating \( \Omega(K) \). The parallelization is implemented by classifying the conformations by the shape of the box enclosing a conformation. Only the conformations corresponding to a given box are enumerated at a time by pruning partial conformations incompatible with the box, and the tasks of enumerating the conformations for the boxes are distributed among computational nodes. Since no communications are required during the calculation, the computational speed scales well with the number of CPUs. The calculation time for each box is reduced by exploiting the symmetries of the system. A conformation, its rotations by multiples of 90°, and their mirror images are considered equivalent and prevented from being generated more than once. For a generic conformation, the discrete rotations and reflections form an 8-fold and 48-fold symmetries in two and three dimensions, respectively. Exceptions are the cases of lower-dimensional conformations embedded in higher-dimensional spaces. In two dimensions, a straight chain is the one-dimensional conformation, invariant with respect to reflection perpendicular to the underlying plane and straight line. Similarly, in three dimensions, only 6-fold and 24-fold symmetries exist for the linear and planar conformations, since they are invariant under transformation perpendicular to the underlying plane and straight line. Therefore, the number of conformations with discrete rotations and reflections considered distinct in \( D \) dimensions, \( \Omega^{(D)}(K) \), can be easily obtained from the reduced number of conformations \( \tilde{\omega}^{(D)}(K) \) where symmetrically related conformations are counted only once and the lower-dimensional conformations are not counted:

\[ \begin{align*}
\Omega^{(1)}(K) &= 2\tilde{\omega}^{(1)}(K) = 2\delta_{K,0}, \\
\Omega^{(2)}(K) &= 8\tilde{\omega}^{(2)}(K) + 4\tilde{\omega}^{(1)}(K), \\
\Omega^{(3)}(K) &= 48\tilde{\omega}^{(3)}(K) + 24\tilde{\omega}^{(2)}(K) + 6\tilde{\omega}^{(1)}(K).
\end{align*} \]

On the other hand, the reduced numbers of conformations \( \tilde{\omega}^{(D)}(K) \) where only symmetries are eliminated, are expressed in terms of \( \omega^{(D)}(K) \) as

\[ \begin{align*}
\omega^{(1)}(K) &= \tilde{\omega}^{(1)}(K) = \delta_{K,0}, \\
\omega^{(2)}(K) &= \tilde{\omega}^{(2)}(K) + \tilde{\omega}^{(1)}(K), \\
\omega^{(3)}(K) &= \tilde{\omega}^{(3)}(K) + \tilde{\omega}^{(2)}(K) + \tilde{\omega}^{(1)}(K).
\end{align*} \]

From Eqs. (7) and (8), we see that \( \Omega^{(D)}(K) \) are expressed in terms of \( \omega^{(D)}(K) \) as:

\[ \begin{align*}
\Omega^{(2)}(K) &= 8\omega^{(2)}(K) - 4\delta_{K,0}, \\
\Omega^{(3)}(K) &= 48\omega^{(3)}(K) - 24\omega^{(2)}(K) - 18\delta_{K,0}.
\end{align*} \]

It is to be noted that the enumeration in two dimensions must be performed before enumerating those in three dimensions.

Although the current algorithm may be used for both homopolymer and heteropolymer in any dimension, as a simple example of the application, we calculate the exact partition function for a homopolymer on a two-dimensional square lattice, for chain lengths up to 36. By analyzing the behavior of the specific heat, we could estimate the temperature of polymer collapse transition.

2. The method

The crucial ingredients in the current method are (1) classifying conformations according to the boxes they span, and enumerating only those for the given box at a time by pruning partial conformations incompatible with the box, and (2) preventing symmetrically related conformations from being generated more than once. The arguments are almost the same for both two and three dimensions, but more detailed illustration will be given for two dimensions which is easier to visualize.

2.1. The classification of the conformations according to the spanning boxes

For each conformation, we construct a rectangular box enclosing it, whose sides are touched. The conformations can then be classified according to the shapes of such boxes, since the box is uniquely determined for each conformation. The shape of a box can be described by the width \( w \) and height \( h \) in two dimensions, and the depth \( d \) is added for three dimensions. We use the convention that these numbers are measured in the unit of the lattice spacing, so the number of lattice sites spanned by the box is \( (w+1) \times (h+1) \times (d+1) \). For simplicity of the discussion, we keep \( d \) regardless of the dimensions, which is 0 for two dimensions. As an illustration, conformations with box sizes 5 × 3 and 4 × 4 in two dimensions are depicted in Fig. 1. There is an intrinsic direction in the polymer chain, and a conformation for \( N \) monomers can be considered as a self-avoiding walk of \( N-1 \) steps, starting from the first monomer.

Since the area or volume of a box should be large enough to accommodate a conformation, \( w, h, \) and \( d \) should satisfy the lower bound \( (w+1) \times (h+1) \times (d+1) \geq N \) for a chain with \( N \) monomers. Also, since the perimeter of the box should be small enough so that all sides touch the conformation, they should also satisfy the upper bound \( w+h+d \leq N-1 \). Due to discrete symmetries, it is enough to consider only the boxes with \( w \geq h \geq d \). Also, the conformations for boxes with \( w, h, \) and \( d \) that saturate the upper bound, \( w+h+d = N-1 \), do not have to be enumerated explicitly, since they can be obtained from a simple analytic formula. Considered as self-avoiding walks, these are the conformations where the steps are taken in a fixed direction along each axis, without turning back, making no intra-chain contact at all. Since there are \( D \) possible directions for each step in \( D \) dimensions, the total number of such conformations for all possible boxes is simply \( D^{N-1}/D! \) when rotational and reflectional symmetries are eliminated. Therefore, the values of \( w, h, \) and \( d \) that have to be included in the explicit enumeration are integers bounded by the following inequalities:

\[ w \geq h \geq d, \]

\[ (w+1) \times (h+1) \times (d+1) \geq N, \]
where explicit enumeration is performed is shown in Fig. 2 for an early stage. This is done by keeping the record of whether each conformation incompatible with the current box is pruned out at enumerating the conformations spanning a given box, any partial that the results from all the nodes are added to obtain enumeration, and it is only after all the computations are finished must be precalculated in order for this efficient distribution, which any of the nodes. The total number of boxes for each chain length to be so efficient since certain nodes finish jobs earlier and become free while others keep enumerating, due to the fact that the number of conformations and the computation time vary widely among the boxes. Therefore, the most efficient way of job distribution is to make enumeration for each box a separate task that can be taken up by any free node. Since a computational node that finishes the enumeration takes over a job in the queue, there is no idle time for any of the nodes. The total number of boxes for each chain length must be precalculated in order for this efficient distribution, which can be performed in a practically negligible amount of time. Absolutely no communication is needed between the nodes during the enumeration, and it is only after all the computations are finished that the results from all the nodes are added to obtain \( \Omega(K) \). The current algorithm can be used in a single CPU as well, by requiring it to enumerate conformations for all possible boxes.

2.2. Self-avoiding walk and pruning

The conformations are enumerated by generating self-avoiding walks. This is most easily achieved by recursively calling a subroutine that makes one step into a given direction [27]. The number of sites \( n \) occupied by the current partial conformation is kept as a global variable, which is equivalent to the number of lattice sites visited by the walk so far. From the given position, the next step is made for each of the neighboring lattice sites not occupied by a monomer. For each step of the walk, the current lattice site is marked as occupied, and the contribution of the current monomer is added to the number of contacts \( K \). When \( n \) reaches \( N \), the contribution of the final monomer is added to \( K \) and \( \omega(K) \) is incremented by one.

In order to ensure that the computation time is spent only for enumerating the conformations spanning a given box, any partial conformation incompatible with the current box is pruned out at an early stage. This is done by keeping the record of whether each side of the box has already been touched by the walk generated so far. If there is a boundary that has not been touched as yet, whose distance from the current position is \( l \), then the next step in the opposite direction to this boundary is forbidden unless there are sufficient number of remaining monomers \( N - n \). Assuming the untouched boundary is at \( x = 0 \) and the current position is at \( x = l \) and the next step is in the positive \( x \) direction, at least 3 monomers are needed to make a \( U \)-turn and get back to \( x = l \), and \( l \) monomers are needed to reach the boundary \( x = 0 \), leading to the inequality

\[
I + 3 \leq N - n, \tag{11}
\]

where \( d \) is set to 0 in two dimensions. The region of \( w \) and \( h \) where explicit enumeration is performed is shown in Fig. 2 for the two-dimensional homopolymer along with the boundaries.

Only the conformations for a given box are enumerated at a time, and parallel computation is performed by distributing the boxes to the computational nodes. Since the number of boxes \( N_{\text{box}} \) is generally larger than the number of computational nodes \( N_{\text{cpu}} \), the simplest method of distribution would be to assign the same number of boxes to each node. However, this method turns out not to be so efficient since certain nodes finish jobs earlier and become free while others keep enumerating, due to the fact that the number of conformations and the computation time vary widely among the boxes. Therefore, the most efficient way of job distribution is to make enumeration for each box a separate task that can be taken up by any free node. Since a computational node that finishes the enumeration takes over a job in the queue, there is no idle time for any of the nodes. The total number of boxes for each chain length must be precalculated in order for this efficient distribution, which can be performed in a practically negligible amount of time. Absolutely no communication is needed between the nodes during the enumeration, and it is only after all the computations are finished that the results from all the nodes are added to obtain \( \Omega(K) \). The current algorithm can be used in a single CPU as well, by requiring it to enumerate conformations for all possible boxes.

2.3. Elimination of discrete symmetries

The speed of enumeration is increased by generating symmetrically related conformations only once. Since the discrete symmetries of rotation and reflection are 8-fold and 48-fold in two and three dimensions respectively, the enumeration time is reduced by nearly the same factor by calculating the reduced number of conformations \( \omega(K) \) instead of the full number \( \Omega(K) \). For a rectangular box where \( w \), \( h \), and \( d \) are all different, 90° rotational symmetry is removed by considering only the box with \( w > h > d \). In order to remove the remaining 4-(8-)fold symmetry in two (three) dimensions, we divide the rectangular box into 4 (8) equivalent quadrants (octants) containing each corner of the box, and consider only the chain that starts inside one quadrant (octant), given by:

\[
1 \leq x \leq \frac{w}{2} + 1, \\
1 \leq y \leq \frac{h}{2} + 1, \\
1 \leq z \leq \frac{d}{2} + 1, 
\tag{13}
\]

where \( x, y, \) and \( z \) are integer coordinates for the lattice sites. The quadrant described by Eq. (13) in the case of two-dimensions is shown as gray area in Fig. 3(a). Since symmetries are not completely eliminated for conformations starting at the boundaries, where any of the conditions, Eq. (13), is satisfied as an equality, additional constraints are imposed, so that the first step along the axis corresponding to an equality is in the positive direction. For example, when \( w \) is an even number and if the starting position of the chain is at \( x = \frac{w}{2} + 1 \), we require that the first step along the \( x \)-axis should be positive. Examples in two dimensions are shown in Figs. 3(b) and (c).

When any two of the three numbers \( w, h, \) and \( d \) are equal, for example \( w = h \), there remains a reflection symmetry with respect to plane \( x = y \), which is eliminated by imposing an additional constraint that among the steps parallel to the \( x-y \) plane, the first one must be along the \( x \) direction. Similarly, when \( h = d \) we break the symmetry by requiring that among the steps parallel to the \( y-z \) plane, the first step must be along the \( y \) axis. Again the two-dimensional example is given in Fig. 3(d).

3. Result

The method developed in the current work can be applied to either heteropolymer or homopolymer in any dimension, but as
a simple example of the application, we study the homopolymer in the two-dimensional square lattice. Since a monomer cannot make a contact with itself, as well as the nearest and next-nearest neighbors along the chain, the upper limit of the $K$ summation in Eq. (5), denoted as $K_{\text{max}}$, satisfies the upper bound

$$K_{\text{max}} \leq \frac{N(N - 5)n_{\text{max}}}{2} + 3n_{\text{max}},$$

(14)

where $n_{\text{max}}$ is the maximum value of $n(a, b)$. In the absence of an additional information, we may take this value as the size of the array to store the values of $\omega(K)$, but for the special case of the homopolymer in two dimensions, we have the exact formula for $K_{\text{max}}$ [28]:

$$K_{\text{max}} = \begin{cases} N - 2m & \text{for } m^2 < N \leq m(m + 1), \\ N - 2m - 1 & \text{for } m(m + 1) < N \leq (m + 1)^2, \end{cases}$$

(15)

where $m$ is a positive integer.

We calculated the number of states $\omega(K)$ for $N \leq 36$. The same quantities for $N \leq 28$ have been calculated in earlier works [1, 28,29]. Our calculation reproduce these results in the appropriate ranges. The new results for $29 \leq N \leq 36$ are presented in Tables 1 and 2. The CPU time using Intel Xeon CPUs (2.8 GHz) is plotted in logarithmic scale as the function of $N$ in Fig. 4, for $15 \leq N \leq 30$ where time data are available. As expected, the CPU time grows exponentially as $N$ increases:

$$t \simeq A\lambda^N$$

(16)

where values of $A = 1.64(2) \times 10^{-8}$ and $\lambda = 2.43(2)$ are obtained by taking the log of Eq. (16) and performing the least square fit. The total number of conformations as well as the computational times of many enumeration algorithms are known to grow exponentially as $\mu^N$ where $\mu = 2.638$ is the connective constant for the self-avoiding walks [30], so the computational time of the current algorithm grows at somewhat slower rate than this. Although algorithms for enumerating the total number of conformations have been developed [31,32] whose computation time grows at rates slower than the current one, it must be noted that the current method not only calculates the total number but also the number of conformations for each energy value, leading the calculation of the exact partition function at arbitrary temperature.

By the efficient distribution of the enumeration tasks among the computational nodes, the computational speed scales linearly with the number of computational nodes until the saturation occurs. It is obvious that the number of nodes used, $N_{\text{CPU}}$, cannot exceed the number of boxes, but the saturation occurs at a smaller value of $N_{\text{CPU}}$ because the enumeration time varies widely among the boxes. A few nodes enumerating the box with large number of conformations tend to keep computing even after most of the nodes have completed computations, causing the deviation from the linear scaling of computation speed with $N_{\text{CPU}}$. By recording the computation times for each boxes and assuming the most efficient distribution of tasks between the nodes, the computation time could be calculated as a function of $N_{\text{CPU}}$. The speed of computation, the inverse of the computation time, is plotted in Fig. 5 for various values of $N$, where the saturated value of the speed is
normalized to one. We see that the linear scaling holds for up to $N_{\text{CPU}} = N_{\text{CPU}}^{(\text{max})}$ which increases with $N$.

As an example of the application of our method, we calculate the specific heat per monomer,

$$C(T, N)/\epsilon^2 N = \frac{1}{\epsilon^2 N} \frac{\partial^2 E}{\partial T^2} = \frac{\rho^2}{\epsilon^2 N} \frac{\partial^2 \ln Z}{\partial \rho^2}$$

$$= \frac{Z}{N} \left[ \sum_k k^2 \Omega(k) z_k \right] \left[ \sum_{\rho} \frac{\rho}{\Omega(\rho)} k^2 z_{\rho} - \left( \sum_{\rho} k \Omega(k) z_{\rho} \right) \right]$$

(17)

which is plotted in Fig. 6 as a function of $z$ for several values of $N$. The finite $N$ approximation of the transition point, $z_c(N)$, is obtained from the condition $\frac{\partial^2 E}{\partial T^2} = 0$. We observe a peak around $z_c(N)$, which becomes sharper as $N$ increases. The point of the collapse transition is known to follow the finite-size scaling

$$z_c(N) - z_c(\infty) \sim N^{-\phi},$$

(18)

where $\phi$ is the crossover exponent whose exact value is believed to be $3/7$ [15]. We apply the Bulirsch–Stoer extrapolation [33–35].
the extrapolated value $z_c$ and its extrapolated value at $N = \infty$.

to Eq. (18) and then obtain $z_c(\infty) = 2.07(7)$, equivalent to $T_c/\epsilon = 1.37(7)$, where the data for even $N$ with $20 \leq N \leq 36$ were used. $z_c(N)$ is displayed in Fig. 7 as the functions of $1/N$, along with the extrapolated value $z_c(\infty)$. The current result is in reasonable agreement with those from the earlier works.

4. Discussion

We developed an efficient parallel algorithm for calculating the exact partition function of a lattice polymer. An efficient parallelization of the calculation was achieved by classifying the conformations according to the shape of the box spanned by a conformation. Only the conformations corresponding to a given box were enumerated at a time, pruning partial conformations incompatible with the box at an early stage. The calculation time was further reduced by preventing the conformations related by symmetries from being generated more than once. The parallel efficiency could be maximized by requiring that any node that finishes the task of enumeration for a box takes over a new task of enumeration for another box whose conformations have not been enumerated by any of the nodes. As an illustration of applications, we studied the collapse transition of lattice homopolymers in square lattices, by calculating the specific heat. The exact partition function can also be used for calculating the partition function zeros in the complex temperature plane, which is a more sensitive indicator of the phase transition than the specific heat [36].

As mentioned in the previous section, the linear scaling of the computation speed with the number of CPUs $N_{CPU}$ breaks down well before $N_{CPU}$ reaches the total number of boxes, due to the fact the computation time varies wildly depending on boxes. However, considering the fact that $N_{CPU}$ cannot be made arbitrarily large in practice, the saturation effect can be neglected in many situations, particularly when long chains are studied, since the saturation limit $N_{CPU}^{(\text{max})}$ grows with the chain length $N$ (see Section 3). As can be seen from Section 3 (see Fig. 5), the partition function of the two-dimensional polymer chain of length 29 can be enumerated with up to 30 CPUs without saturation of the linear scaling, and one can employ more CPUs for studying chains of longer lengths. Since linear scaling becomes better as the chain length increases, our method is a powerful tool for studying long polymer chains with limited computational resources.

An idea of classifying conformations of two-dimensional homopolymers according to spanning boxes, similar to ours in certain aspects, has been introduced for the calculation of the partition function of a polymer on the square lattice at infinite temperature using transfer matrix formalism [32]. At least for such a calculation the transfer matrix was claimed to be superior to the direct counting, and it would be interesting to see whether it can be generalized for calculating the partition function at arbitrary temperatures in two dimensions, without introducing much additional computational costs. The current algorithm is more general in that not only the partition function at arbitrary temperature can be calculated, but heteropolymers and arbitrary dimensions can be treated in a straightforward manner. The explicit applications of the algorithm to these cases are left for future works.

Acknowledgements

This work was supported by Mid-career Researcher Program through NRF grant funded by the MEST (No. 2010-0000220).

References