NONREVERSIBLE MARKOV CHAIN MONTE CARLO ALGORITHM FOR EFFICIENT GENERATION OF SELF-AVOIDING WALKS

Hanqing Zhao

Beijing, China

Bachelor of Science, Wuhan University, 2019

A Thesis submitted to the Graduate Faculty
of the University of Virginia in Candidacy for the Degree of
Master of Science

Department of Physics

University of Virginia

May 2022

Advisor M. Vucelja, Chair

Second G. Chern

Nonreversible Markov chain Monte Carlo algorithm for efficient generation of Self-Avoiding Walks

Hanqing Zhao

(ABSTRACT)

A Self-Avioding Walk (SAW) is defined as a contiguous sequence of moves on a lattice that does not cross itself. Typically one uses Monte Carlo approaches (A. Sokal 1997; Newman and Barkema 1999) to generate SAW numerically. We introduce an efficient nonreversible Markov chain Monte Carlo algorithm to generate self-avoiding walks with a variable endpoint. In two dimensions, the new algorithm slightly outperforms the *two-move nonreversible Berretti-Sokal algorithm* (Hu, X. Chen, and Deng 2016), while for three-dimensional walks, it is 3–5 times faster. The new algorithm introduces nonreversible Markov chains that obey global balance and allow for three types of elementary moves on the existing self-avoiding walk: shorten, extend or alter conformation without changing the length of the walk.

Dedication

To everyone who has helped me throughout my education.

Acknowledgments

First I would like to thank my supervisor Marija Vucelja for her kind and effective instructions on both my coursework and research. It has been a wonderful time working with her. Secondly I would also like to thank my department giving me this valuable opportunity and supporting me in the past semesters. I would also like to thank my parents. And finally I thank the admission committee and the professors who taught me in the past semesters.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the most fundamental problem in polymer physics is the simulation and enumeration of polymer conformations, which poses many interesting and challenging questions in areas including statistics and statistical mechanics. For a linear polymer, the simplest model is Monte Carlo simulation modelled by the self-avoiding walk which will be discussed in detail.

Self-avoiding walks have been studied since the 1940s and great theoretical breakthroughs have been made by the use of probability theory, rigorous constructive techniques, scaling arguments and conformal field theory (P. J. Flory 1949). Monte Carlo simulations can be used to generate self-avoiding walks and are a collection of versatile and robust algorithm. Many of these algorithms can also be used in more general models of walks like the interacting walks which are usually used as models of interacting polymers.

A Self-Avoiding Walk (SAW) is defined as a contiguous sequence of moves on a lattice that does not cross itself; it does not visit the same point more than once. SAWs are fractals with fractal dimension 4/3 in two dimensions, close to 5/3 in three dimensions, and 2 in dimensions above four (Havlin and Ben-Avraham 1982; S. Havlin and D. Ben-Avraham 1982). In particular two-dimensional SAWs are conjectured to be the scaling limit of a family of random planar curves given by the Schramm-Loewner evolution with parameter $\kappa = 8/3$ (Lawler, Schramm, and Werner 2002). Since their introduction, SAWs have been used to model linear polymers (P. Flory 1953; Metropolis and Ulam 1949; Rensburg 2009). They are essential for studies of polymer enumeration where scaling theory, numerical approaches, and field theory are too hard to analyse (de Carvalho, Caracciolo, and Frohlich 1983; Duplantier 1986). SAWs are also used in the numerical studies of finite-scaling (Zhou, Grimm, Fang, et al. 2018) and two-point functions (Zhou, Grimm, Deng, et al. 2020) of Ising model and $n-$vector spin model (Fang, Deng, and Zhou 2021). Analytical results on SAWs are scarce, and generating long SAWs is computationally complex.

Typically one uses Monte Carlo approaches (A. Sokal 1997; Newman and Barkema 1999) to generate SAWs numerically. Many previous Markov chain Monte Carlo (MCMC) algorithms have been designed to efficiently produce different kinds of SAWs by manipulating potential constructions that can be executed on a walk to increase, decrease its length, or change its conformation. For example, the pivot algorithm samples fixed-length SAWs – it alters the walk's shape without changing its length (Madras and A. D. Sokal 1988). While the Berretti-Sokal algorithm and BFACF algorithm contain length-changing moves and can generate walks with varying lengths (Berretti and A. D. Sokal 1985; Rensburg and Whittington 1991). Here we have discussed the definition of self-avoiding walks and in the next few sections we will discuss some interesting applications and also studies of self-avoiding walks in detail.

1.1 Application of Self-Avoiding Walks

Traditionally, self-avoiding walks have been used to model structural and dynamical properties of macromolecules (de Gennes and Witten 1980). For example, self-avoiding walks confined to clusters of the percolation problem on $2D$ and $3D$ lattices is a direct analog of the problem of a linear-chain polymer trapped in a porous medium where excluded regions can occur with the length scales of the order of the persistence length of the chain. Some researches also use self-avoiding walks to study the critical phenomena in lattice models (Lee, Nakanishi, and Kim 1989). By using a Monte Carlo study of self-avoiding walks on a diamond lattice, the researches conducted an extensive study of the behaviour of linear polymer chains on diamond lattices in the region around the θ-point. In this study, the model of a polymer consists of N bonds of fixed lengths on a diamond lattice which has the coordination number $z = 4$. And either simple sampling or importance sampling can be used.

Additionally, self-avoiding walks serve as suitable tools to probe the large-scale topological structure of complex networks. The self-avoiding walks are expected to be more suitable than unrestricted random walks to explore various kinds of real-life networks since they cannot return to sites already visited. This property has been used to define local search strategies in scale-free networks (Adamic et al. 2001). And self-avoiding walks on networks may be used to describe agents or robots propagating and damaging a network of computers, such that damaged nodes are effectively wiped out from the network. Meanwhile, finding communities in networks have great and practical importance in metabolic process, marketing strategies and improving the routing in World Wide Web. The community detection in complex networks has high computational complexity of the optimization process in a network, which could be advantageous to use the high effectiveness of self-avoiding walks. Researches have used a SAW-based method to extract the community distribution of a network and found that it achieved high modularity scores, especially for real-world networks (de Guzzi Bagnato, Ronqui, and Travieso 2018). It is noted that the self-avoiding property causes attrition of the paths where a large fraction of paths generated in a stochastic manner have to be abandoned because they are overlapping. However, more efficient algorithms can be used to overcome this serious limitation to explore networks with self-avoiding walks.

Recently, studies also use the MCMC algorithms to generate self-avoiding walks in order to study the two-point function of Ising model (Zhou, Grimm, Fang, et al. 2018; Zhou, Grimm, Deng, et al. 2020; Fang, Deng, and Zhou 2021), which will be discussed in detail later.

1.2 Studies of Self-Avoiding Walks

For studies of SAWs, one of the most fundamental quantities is c_n, which is the number of walks of length n starting from the origin. It is easy to determine the value of c_n when the length is small. For example, we have $c_0 = 1$ and $c_1 = 2d$ where d is the dimension. However, when the length grows, it quickly becomes very difficult to determine c_n. Another interesting value is the growing constant μ which is defined as $\mu = \lim_{n\to+\infty} \frac{c_{n+1}}{c_n}$. In the past, exact enumeration and series analysis of walks as well as Monte Carlo simulations have been used to estimate the two variables in different systems. In general, a numerical approach to the self-avoiding walk can be implemented to verify results obtained by other means, or to determine some variables associated with the model. When a Monte Carlo algorithm is use, one of the greatest questions is how to sample walks efficiently according to the need. Therefore, in the next chapter, we are going to present some famous Monte Carlo algorithms which are used to generate SAWs under different circumstances.

Chapter 2

Previous Research and Work

Monte Carlo method is a well-known method for sampling a statistical distribution. Its implementation via the Metropolis algorithm was first invented in 1953. The implementation of Monte Carlo algorithm to generate walks is the sampling of walks from a distribution over a state space. And the estimation of expected values off observables can be carried out after it.

One of the research interests in this area is the invention of new Monte Carlo algorithms for sampling self-avoiding walks (Rensburg 2009). Since the Rosenbluth method's invention in 1955, a great number of new methods have been invented including new algorithms and several dynamic algorithms which can be used to simulate the dynamics of lattice polymer chains (M. N. Rosenbluth and A. W. Rosenbluth 1955). Several new approaches have been designed since the 1980s including the BFACF algorithm, the Berretti-Sokal algorithm and the pivot algorithm (Rensburg and Whittington 1991; Berretti and A. D. Sokal 1985; Madras and A. D. Sokal 1988). By manipulating different types of atmospheres, these algorithms can be used to sample self-avoiding walks under different conditions. For example, the Beretti-Sokal algorithm can generate self-avoiding walks with an unfixed endpoint and different lengths via manipulating the endpoint atmospheres. The BFACF algorithm, on the other hand, can sample self-avoiding walks with changing lengths but a fixed endpoint. At the same time, the Rosenbluth method has been modified into PERM and

GARM algorithm, which are static Monte Carlo algorithms based on simple sampling but with ingenious additions to improve sampling such as pruning and enrichment (Rechnitzer and Rensburg 2008; Hsu and Grassberger 2004; Owczarek and Prellberg 2001).

In general, the implementation of a particular Monte Carlo algorithm involves with manipulating different types of atmospheric moves. And the Berretti-Sokal algotithm is one of the state-of-art algorithm as it could generate SAWs with different endpoints and lengths which can have applications in many aspects. By using the nonreversible techniques and including all endpoint atmoshperic moves, we have achieved higher efficiency in $2D$ and $3D$ systems comparing to previous BS type algorithms. Therefore, in the next chapter, we will discuss atmosphere and atmospheric moves as well as the orginal Berretti Sokal algorithm.

Chapter 3

The Berretti-Sokal Algorithm

The Berretti-Sokal algorithm manipulates the endpoint atmospheres so this chapter begins with the introduction of atmospheres and atmospheric moves. Then before the introduction of the Berretti-Sokal Algorithm, we will discuss the balance condition which is the most important factor in designing an MCMC algorithm since it ensures that the convergence of the Markov chain. Note that the probability distribution of a SAW of length $|s|$ is

$$\pi \propto x^{|s|} \tag{3.1}$$

where x is the weight of a unit step. This is what we want the Markov chain target distribution to be.

3.1 The Atmospheres

The algorithms creating SAWs usually manipulate different kinds of proposed moves, often referred to as *atmospheres* (Rensburg and Rechnitzer 2008; Rechnitzer and Rensburg 2002; Rensburg and Rechnitzer 2009). Atmospheres can be described as potential constructions that can be executed on a given walk to increase or decrease the current length or change the conformation. When generating SAWs, the algorithm usually performs moves on either endpoint atmospheres or plaquette atmospheres where *positive* and *negative* atmospheres are generally defined as ways of adding or removing a fixed number of edges to the current walk. In contrast, *neutral* moves are ways of altering the walk's shape without changing its length. For instance, the pivot algorithm, which only acts on neutral atmospheres, can be used to sample fixed-length walks (Madras and A. D. Sokal 1988). In contrast, the Berretti-Sokal algorithm and BFACF algorithm contain length-changing atmospheric moves and can generate walks of different lengths (Berretti and A. D. Sokal 1985; Rensburg and Whittington 1991).

In general, algorithms generating SAWs are manipulating either the *endpoint atmospheres* or the *plaquette atmospheres*. The definitions and examples are shown as follow.

For the *endpoint atmosphere*, suppose s is the current SAW starting from the origin with length $|s|$ and its last vertex is v. The positive endpoint atmospheres are the lattice edges incident with the last vertex, which can be occupied to extend the length by one. The negative endpoint atmosphere is just the last occupied edge since removing it can extract the length by one. The neutral endpoint atmospheres are edges that can be occupied by changing the direction of the vertex v. For any SAW with a non-zero length, the number of negative endpoint atmospheres is one. If the SAW has zero length, the number of negative endpoint atmospheres is set to zero, as the length can not be further reduced.

Fig. 3.1 shows a SAW with a length equal to four. In this example, three unoccupied edges are incident with the last vertex; they are shown in red on the graph, making three positive ending atmospheres. As we see from the last occupied edge (black arrow), there is just one negative endpoint atmosphere. There are two neutral endpoint atmospheres, and the corresponding edges are displayed with green arrows.

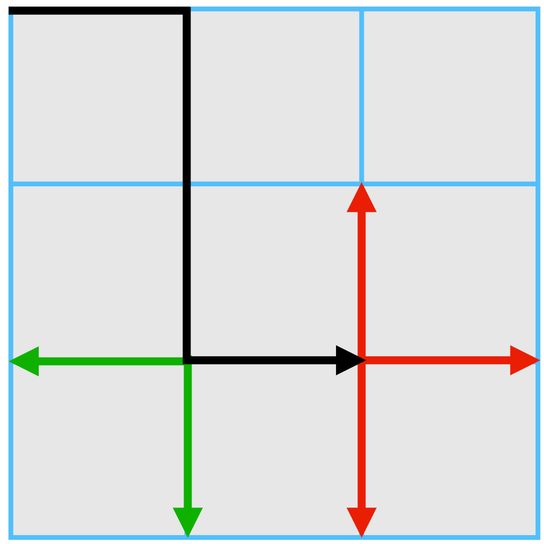

Figure 3.1: The endpoint atmospheres on a self-avoiding walk of length $|s| = 4$. For this self-avoiding walk, there are three positive ending atmospheres (red arrows) and one endpoint atmosphere, which is the last occupied edge (black arrow), and the number of neutral endpoint atmospheres is two (green arrows).

For the *plaquette atmosphere*, if s us an SAW of length $|s|$ from the origin, then three successive edges in a $\sqcup$-conformation is a negative plaquette atmosphere. Conversely, if an edge in s can be replaced by three edges in a $\sqcup$-conformation to create a new SAW of length $|s| + 2$, then the edges form a positive plaquette atmosphere. Additionally, two adjacent edges incident at 90° with one another and bounding a unit square with exactly two edges and three vertices in the walk is a neutral plaquette atmosphere. Fig. 3.2 shows an example of platuette atmospheres in a SAW with a length equal to nine.

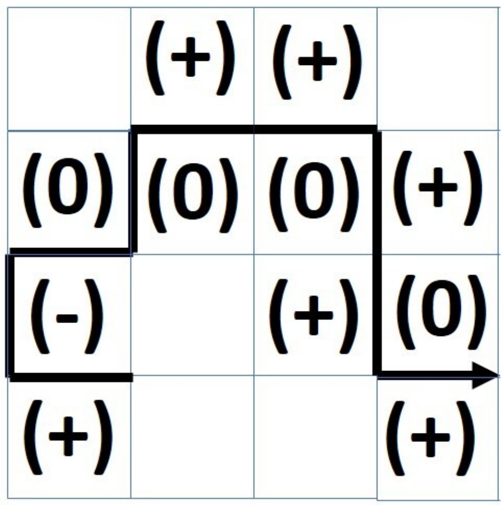

Figure 3.2: The plaquette atmospheres on a self-avoiding walk of length $|s| = 9$. The neutral plaquette atmospheres are shown in (0) while the positive and negative plaquette atmospheres are shown in (+) and (−). For this self-avoiding walk, there are six positive plaquette atmospheres and one negative plaquette atmosphere, and the number of neutral plaquette atmospheres is four.

3.2 Detailed Balance and Global Balance

The balance condition is one of the most important factors in designing an MCMC algorithm since it ensures that the Markov chain will converge to a target distribution. The above described MCMC algorithms satisfy the detailed balance condition - which states that the weighted probabilities of transitions between states are equal. For the detailed balance condition (DBC) we have

$$P_{ij}\pi_j = P_{ji}\pi_i, \quad \forall i, j \in \Omega, \tag{3.2}$$

where P_{ij} is the transition probability from state j to state i, Ω is the space of states, and π is the stationary distribution, see e.g. (Levin, Peres, and Wilmer 2009; Vucelja 2016). Detailed balance is a local condition and thus easy to implement. MCMC algorithms using the detailed balance condition use reversible Markov chains. The reversibility introduces a diffusion-like behavior in the space of states.

However, for a Markov chain to asymptotically converge to a stationary distribution π, all we need is a weaker condition – the Global Balance Condition (GBC):

$$\sum_{j \in \Omega} P_{ij}\pi_j = \sum_{j \in \Omega} P_{ji}\pi_i, \quad \forall i \in \Omega, \tag{3.3}$$

where Ω is a space of states. The GBC physically means that the total probability influx at a state equals the total probability efflux from that state (Turitsyn, Chertkov, and Vucelja 2011; A. Sokal 1997). In recent years, there has been progress in designing nonreversible Markov chains that converge to the correct target distribution. Such chains due to ”inertia” reduce the diffusive behavior, sometimes leading to better convergence and mixing properties compared to the reversible chains (Diaconis, Holmes, and Neal 1997; F. Chen, Lovasz, and Pak 1999; Turitsyn, Chertkov, and Vucelja 2011; Vucelja 2016; Sakai and Hukushima 2013; Joris Bierkens and Roberts 2017; J. Bierkens 2016; Kapfer and Krauth 2017).

3.3 Reversible Berretti-Sokal Algorithm

The Berretti-Sokal algorithm is one of the most famous reversible MCMC algorithms which manipulate the endpoint atmospheres. The algorithm only considers the positive and negative endpoint atmospheres and thus has the increasing and decreasing move. Here we are using a Metropolis-Hastings style (Metropolis, A. Rosenbluth, et al. 1953; Hastings 1970) implementation of the Berretti-Sokal algorithm. It works as follows:

(i) Suppose the current length of a SAW is given by N. With equal probability, the algorithm chooses the increasing move or the decreasing move.

(ii) If the increasing move is selected, with probability P_+ one of the empty edges incident with v_N, the last vertex, will be occupied randomly when this leads to a valid SAW of $N+1$ steps. Similarly, for the decreasing move, the last occupied edge is deleted with probability P_-. The two probabilities are given by

$$P_+ = \min\{1, x(z-1)\}, \tag{3.4}$$

$$P_- = \min\left\{1, \frac{1}{x(z-1)}\right\}, \tag{3.5}$$

where z is the coordination number of the system, i.e. the number of lattice points neighboring a vertex on the lattice.

The moves are executed only if they lead to a valid SAW. Special attention is needed for the "null" walk, $|s| = 0$, in such case only an increasing mode is allowed and the number of empty edges is z, rather than $z-1$. For simplicity we permanently set $P_+ = \min\{1, x(z-1)\}$. To prove that the DBC holds in the Berretti-Sokal algorithm, let us for example consider the case where $x(z-1) < 1$. From Eqs. (3.4) and (3.5) we conclude that the choice implies $P_+ < 1$ and $P_- = 1$. Thus we have $x^{|s|}P_+(z-1)^{-1} = x^{|s+1|} = x^{|s+1|}P_-$, which satisfies the DBC, given in Eq. (3.2). The proof is analogous in the case $x(z-1) > 1$.

The Berretti-Sokal algorithm is especially useful for long chains. But in MCMC algorithms, one of the greatest question is how to further improve its efficiency. The Berretti-Sokal algorithm satisfies the detailed balance condition which is a sufficient but not necessary condition to ensure the MCMC algorithm will converge to the target distribution. So it is a reversible MCMC algorithm. It is noted that fast mixing is hindered in traditional reversible MCMC methods due to high barriers in effective energy landscape and high entropy of the states basin. Meanwhile reversible Markov chain Monte Carlo algorithms are especially slow close to phase transitions. Therefore, possible improvement could be made by breaking the detailed balance condition, which may lead to higher efficiency. In the next chapter we are going to discuss how to break the detailed balance of the Berretti-Sokal Algorithm and also its results.

Chapter 4

The Nonreversible Berretti-Sokal Algorithm

In the previous chapter, we have introduced the detailed balance and global balance condition. Generally speaking, the global balance condition is more general though it could be difficult to be implemented due to the great number of possible probability flows. As detailed balance is a special case of global balance, people turned to use global balance condition to design MCMC algorithms which are nonreverisble MCMC algorithms. Most nonreversible MCMC algorithms need to be designed carefully and one of the most appealing ideas is lifting which can increase the phase space to create a bias and explore the enlarged phase space more efficiently (Turitsyn, Chertkov, and Vucelja 2011; Vucelja 2016). It can be carried out by adding a number of subsystems (replicas) with internal dynamics, each characterized by its own transition matrix. Nonreversible MCMC algorithms now are designed carefully and artificially and there is no general rule for all algorithms, which could put great difficulties when designing comparing to reversible MCMC algorithms. Additionally though lifting could alter the convergence time, it is still an open question whether and when it will decrease the convergence time. For algorithms sampling self-avoiding walks, only the Berretti-Sokal algorithm has been modified into a nonreversible MCMC algorithm (Hu, X. Chen, and Deng 2016) which we will call *two-move nonreversible Berretti-*

Sokal algorithm. Besides introducing the nonreversible MCMC technique into the Berretti-Sokal algorithm, we also notice that the previous Berretti-Sokal algorithms have only included the positive and negative endpoint atmosphere moves. So we could include all three types of endpoint atmosphere moves. In this chapter, we will first discuss how our algorithm works and then we will present its performance.

4.1 Nonreversible Berretti-Sokal Algorithm

One possible way to set up a nonreversible algorithm is to increase the phase space by introducing replicas (Turitsyn, Chertkov, and Vucelja 2011; Hu, X. Chen, and Deng 2016; Vucelja 2016) and work on the extended space with nonzero probability fluxes. Here we follow an analogous approach. As mentioned above, there has been a successful *two-move nonreversible Berretti-Sokal algorithm* (Hu, X. Chen, and Deng 2016). The authors achieved an important improvement in the speed of the algorithm. The speedup is about tenfold in two-dimensional systems and is even more pronounced in higher-dimensional systems. They set up two modes in the algorithm, which we call the increasing mode and the decreasing mode.

Our new algorithm has a third type of move – besides shortening and extending the SAW, we also allow the SAW to change its conformation. Namely, in the increasing mode, the algorithm can perform either an increasing move or a neutral move; in this mode, the decreasing move is not allowed. Analogously, in the decreasing mode, the algorithm will only execute either a decreasing move or a neutral move. A diagram describing the algorithm is shown in Fig. 4.1. It works as follows:

i) In the increasing mode, with equal probability, perform either the *positive move* or the *neutral move*. For the *positive move*, the algorithm will randomly occupy

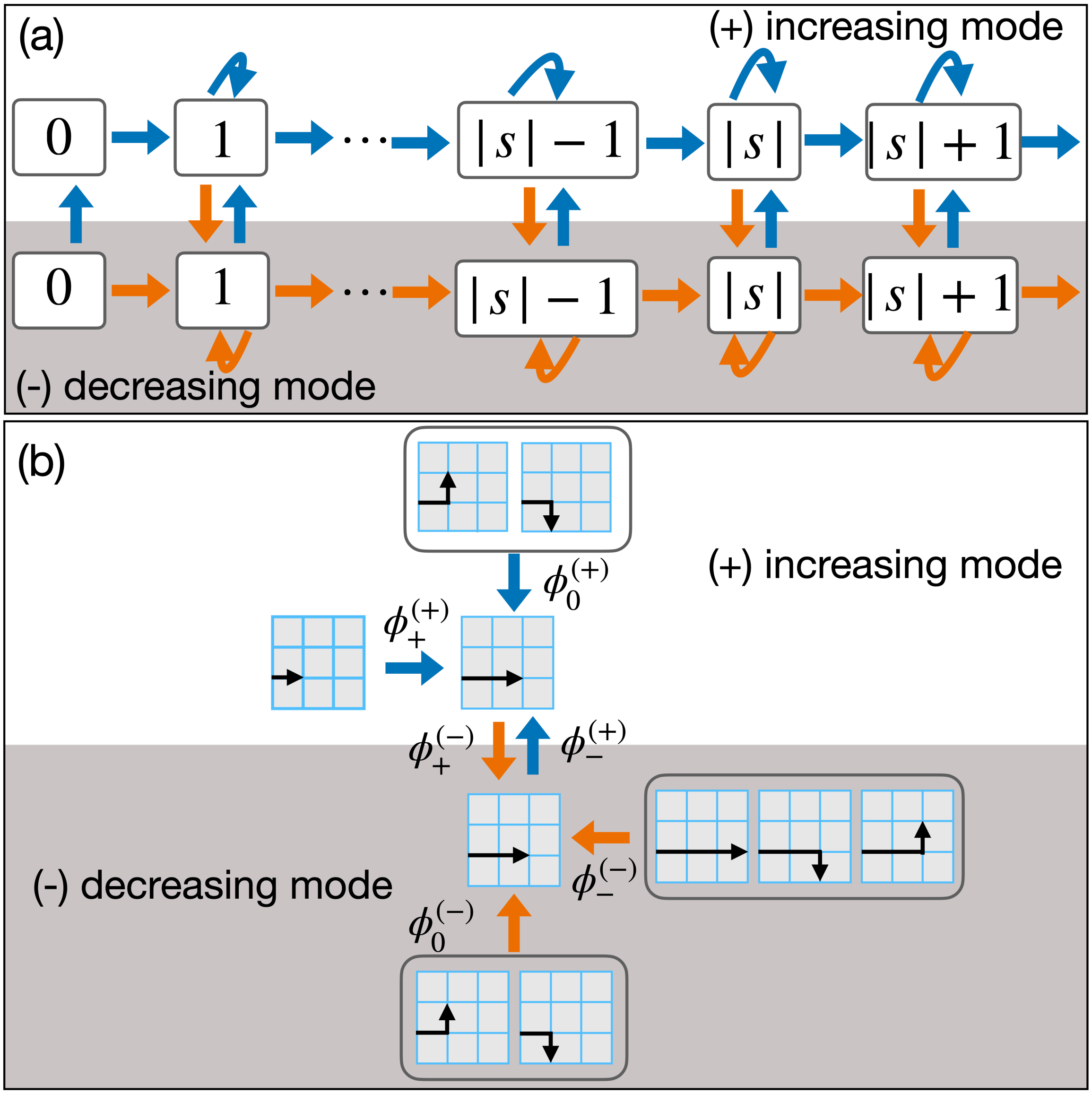

Figure 4.1: **(a)** Diagram of probability flows in the *three-move nonreversible Berretti-Sokal algorithm*. Each rectangle specifies a SAW of length $|s|$. Each realization of the algorithm is different because of the neutral moves, allowing to alter the configuration of the walk. The top row represents the *increasing mode* in which the algorithm can produce either a positive or neutral move, while the bottom row represents the *decreasing mode* where the algorithm produces either negative or neutral moves. The circular arrow represents the execution of a neutral move, leading to a SAW with the same length but a different shape as the last occupied edge's direction is changed. The 'null' walk, $|s| = 0$, requires special attention; in this case, we do not allow neutral and decreasing moves. **(b)** Example of the incoming fluxes for SAW of length $|s| = 2$ in $2D$ on a square lattice.

one of the empty edges incident to the last vertex with probability P_+. While for the *neutral move*, the algorithm will change the direction of its last occupied edge randomly. If the chosen move does not lead to a valid SAW, the algorithm will change to the decreasing mode.

ii) In the decreasing mode, with equal probability, perform either the *negative move* or the *neutral move*. For the negative move, the algorithm will delete the last occupied edge with probability P_-. For the *neutral move*, the algorithm will change the direction of its last occupied edge randomly. If the chosen move does not lead to a valid SAW, the algorithm will change into the increasing mode.

iii) When the length is 0, the algorithm will be changed into the increasing mode, and a *positive move* will be performed.

Therefore, in each step, the algorithm will either execute one of the elementary moves successfully or change to the other mode. The global balance condition implies that the total influx probability flow equals the efflux probability flow; that is, we have

$$\phi_{\pm}^{(\pm)} + \phi_0^{(\pm)} + \phi_{\mp}^{(\pm)} = x^{|s|}, \tag{4.1}$$

where $x^{|s|}$ is the distribution of SAWs of length $|s|$ and $\phi-$s describe the incoming probability fluxes, where the superscript denotes the mode and the subscripts denote the move. The three terms on LHS are the incoming flow of executing a $\pm$ move in mode $(\pm)$, $\phi_{\pm}^{(\pm)}$, the incoming flow of executing one neutral move in mode $(\pm)$, $\phi_0^{(\pm)}$, and the incoming flow from switching the mode from $(\mp)$ to $(\pm)$, $\phi_{\mp}^{(\pm)}$. To clarify the third term in the LHS by example: $\phi_{-}^{(+)}$ is the incoming flux from switching from $(-)$ mode to the $(+)$ mode. Let us show that global balance condition holds for the *increasing mode* when $x(z-1) < 1$. Proofs for the other cases follow analogously. In this case the three fluxes are:

- The incoming flux from a positive move is

$$\phi_{+}^{(+)} = x^{|s|-1} P_{+} \frac{1}{2(z-1)} = \frac{x^{|s|}}{2}, \tag{4.2}$$

  where in the second equality we used Eq. (3.4). The factor 1/2 is the result of selecting either a positive move or a neutral move and the term $(z-1)^{-1}$ is from occupying one of the $z-1$ empty edges incident to the last vertex.

- The incoming flux from a neutral move is

$$\phi_{0}^{(+)} = \frac{x^{|s|} z''}{2(z-1)}, \tag{4.3}$$

  where z'' is the number of possible edges which will lead to a valid SAW for the last occupied edge when changing its direction.

- The incoming flux from the decreasing mode, $\phi_{-}^{(+)}$, since $P_{-} = 1$, as we assume that $x(z-1) < 1$, the only possible reason of changing from another mode is that when the last occupied changes it direction, it does not lead to a valid SAW, thus

$$\phi_{-}^{(+)} = \frac{1}{2} x^{|s|} \left(1 - \frac{z''}{z-1}\right). \tag{4.4}$$

Summing over the incoming flows, given in Eqs. (4.2 - 4.4), we verify that the global balance condition, Eq. (4.1), holds. Note that we do not assume that a particular SAW configuration of length $|s|$ is achieved with the same frequency in the increasing and the decreasing mode – it comes out as a corollary of the global balance condition.

4.2 Result

To test the efficiency of the new algorithm, we used the integrated autocorrelation time τ. For a given observable $\mathcal{O}$, it is defined as

$$\tau = \frac{m}{2}\frac{\sigma^2_{\overline{\mathcal{O}}}}{\sigma^2_{\mathcal{O}}}, \tag{4.5}$$

where m is the number of steps, $\overline{\mathcal{O}}$ is the estimator of the average $\mathcal{O}$, and σ^2 denotes a variance, c.f. (Goodman and Weare 2010). Here we choose the length of the walk, $|s|$, for the observable as it is a common choice for SAWs. We tested the efficiency as a function of the linear system size by generating SAWs in a square lattice with $n \times n$ points and in a cubic lattice with $n \times n \times n$ points. The boundary conditions were fixed. With τ_0 we denote the integrated autocorrelation time of the *two-move nonreversible Berretti-Sokal algorithm* (algorithm from Hu, X. Chen, and Deng 2016).The comparison of the two algorithms is on Fig. 4.2.

Note, that there are two different scenarios based on the value of weight of a unit step x. For example, for a $2D$ square lattice, when $x = 0.4$, $P_+ = 1$ and $P_- < 1$, while for $x = 0.2$, $P_- = 1$ and $P_+ < 1$. To study both scenarios present the results under initial setting where $x = 0.2$ and $x = 0.4$ in a $2D$ system and correspondingly $x = 0.12$ and $x = 0.24$ in a $3D$ system. From Fig. 4.2 we conclude that the ratio of the autocorrelation times for large systems is weakly dependent on the value of x.

In $2D$, the ratio of the autocorrelation time of the new algorithm over the previous one is always less than one, which means that the new algorithm has a slightly

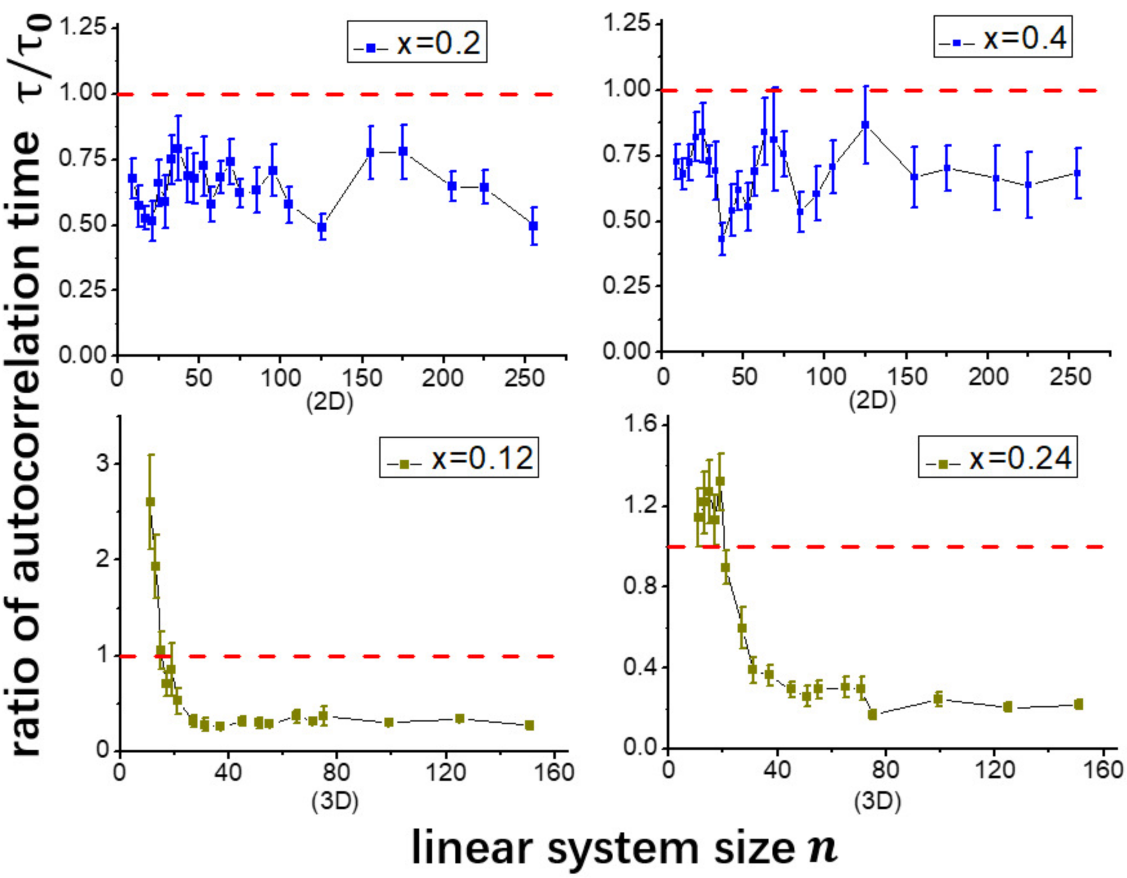

Figure 4.2: The ratio of integrated autocorrelation times of the *three-move nonreversible Berretti-Sokal algorithm*, τ, and the *two-move nonreversible Berretti-Sokal algorithm*, τ_0, for $2D$ and $3D$ systems as a function of the linear system size n. The *three-move nonreversible Berretti-Sokal algorithm*'s performance is slightly better in $2D$ systems while it is $3-5$ times faster in most $3D$ systems.

better performance. We further tested the new algorithm in a three-dimensional cubic system. The new algorithm tends to have better performance in large systems, and the difference is more significant than the $2D$ situation. When the length of the cube is less than 20, the previous algorithm is more efficient with less autocorrelation time. However, as the system's scale increases, the ratio τ/τ_0 becomes less than one, and the value is between 0.2 and 0.3, indicating that the new algorithm is 3 to 5 times faster in these larger $3D$ systems. We have also tested our algorithm in $4D$ and $5D$ systems where no general improvements are found compared to the *two-move nonreversible Berretti-Sokal algorithm.* We show the detailed findings in Appendix A. The fact that the addition of neutral moves does not improve the efficiency in generating SAWs in $4D$ and $5D$, could be explained by the fact that as dimension gets higher, it will be much more likely for the algorithm to make a successful, positive move, which results in less benefit from adding the neutral move.

To summarize, we have created a new nonreversible algorithm manipulating the endpoint atmospheres to generate SAWs. By introducing all three kinds of endpoint atmospheres' moves, the new algorithm has greater flexibility than the *two-move nonreversible Berretti-Sokal algorithm*, from (Hu, X. Chen, and Deng 2016). For instance, when occupied lengths surround the endpoint of a given SAW, the algorithm will change into the negative mode since neither a neutral move nor a positive move will lead to a valid SAW. Assume that $P_+ < 1$, for an algorithm with only positive and negative moves, it will return to the origin and start from the beginning again. On the other hand, with a neutral move, the SAW does not have to start from the origin again. When a neutral move in the negative mode is not possible, the algorithm will change into the positive mode. The addition of neutral moves gives the algorithm greater flexibility in finding valid SAWs.

Chapter 5

Discussion and Conclusions

5.1 Application of Berretti-Sokal Algorithm

As mentioned in the previous chapter, SAWs can be useful tool in the studies of polymer physics, network systems and Ising model as well. At the very beginning of the invention of the Berretti-Sokal algorithm, it was used to verify the growing constant of SAWs, which is one of the traditional applications of SAWs. Nowadays, researchers have found out some other interesting applications of Berretti-Sokal algorithm.

Recently, studies also use the MCMC algorithms to generate self-avoiding walks in order to study the two-point function of Ising model. The two-point function of self-avoiding walk on a finite box T_L^d can be written as

$$g_{SAW}(x) = \sum_{s:0\to x} z^{|s|}, \tag{5.1}$$

where z is the fugacity and the sum is over all self-avoiding walks starting at the origin 0 and ending at x.

For the zero-field ferromagnetic Ising model on T_L^d, the Hamiltonian can be written as

$$H = -\sum_{ij \in E} s_i s_j \tag{5.2}$$

where $s_i \in \{-1, +1\}$ denotes the spin at position T_L^d, and E is the edge set of T_L^d. And its two-point function is

$$g_{Ising}(x) = \langle s_0 s_x \rangle \tag{5.3}$$

which is the expectation value with respect to respect to the Gibbs measure.

By using a resummation of the high-temperature expansion of the Ising model, the Ising two-point function can be written as a weighted sum over edge self-avoiding paths from 0 to x (Thompson 2015). There are multiple ways to construct such paths for a given high-temperature configuration (Aizenman 1986). Meanwhile, the susceptibility for the Ising and SAW models can be written as

$$\chi_{Ising,SAW} := \sum_x g_{Ising,SAW}(x) \tag{5.4}$$

Therefore, the Berretti-Sokal algorithm has been used in researches related to the finite-size scaling of the Ising model in high dimensions by generating self-avoiding walks on a box (Zhou, Grimm, Fang, et al. 2018; Zhou, Grimm, Deng, et al. 2020; Fang, Deng, and Zhou 2021). Simulation results obtained by extensive Monte Carlo simulations of the Ising model and self-avoiding walk have supported researcher's conjecture that on tori of dimension at least 5, the two-point functions of the Ising model and self-avoiding walk display the same scaling as the random-length random walk.

5.2 Future Research Aspects

The number of nonreversible versions of algorithms generating self-avoiding walks is still limited. As the conformation of a given walk could change dramatically after one Monte Carlo step, it is hard to figure all the possible probability flows in the global balance condition. There has been no nonreversible versions of the BFACF algorithm and the pivot algorithm yet. Future research could investigate how atmosphere moves could change the conformation of a SAW thoroughly and then implement the nonvreversible techniques. It would also be interesting to study the possibility of implementing the nonreversible technique in the PERM and GARM algorithm. Previous research has improved the efficiency of PERM algorithm without implementing the nonreversible MCMC techniques Campbell and Rensburg 2020.

Currently, the nonreversible technique is hard to be implemented as it needs to be designed carefully and artificially to satisfy the global balance condition, which hinders its wide implementation. Looking into the future, one might delegate this task to a neural network alike in Song, Zhao, and Ermon 2017. Optimizing the transition operator with more than three types of endpoint atmospheres might further increase the efficacy.

Bibliography

Flory, Paul J. (1949). “The Configuration of Real Polymer Chains”. In: *The Journal of Chemical Physics* 17.3, pp. 303–310. DOI: 10.1063/1.1747243. eprint: https://doi.org/10.1063/1.1747243. URL: https://doi.org/10.1063/1.1747243.

Metropolis, N. and S. Ulam (Sept. 1949). “The Monte Carlo method”. In: *J Am Stat Assoc* 44.247, pp. 335–341.

Flory, Paul (1953). *Principles of Polymer Chemisty*. Cornell University Press.

Metropolis, N., A.W. Rosenbluth, et al. (1953). “Equations of State Calculations by Fast Computing Machines”. In: *J. of Chem. Phys.* 21, pp. 1087–1092.

Rosenbluth, Marshall N. and Arianna W. Rosenbluth (1955). “Monte Carlo Calculation of the Average Extension of Molecular Chains”. In: *The Journal of Chemical Physics* 23.2, pp. 356–359. DOI: 10.1063/1.1741967. eprint: https://doi.org/10.1063/1.1741967. URL: https://doi.org/10.1063/1.1741967.

Hastings, W. K. (1970). “Monte Carlo sampling methods using Markov chains and their applications”. In: *Biometrika* 57, pp. 97–109.

de Gennes, P. G. and Thomas A. Witten (Jan. 1980). *Scaling Concepts in Polymer Physics*. Vol. 33. 6, p. 51. DOI: 10.1063/1.2914118.

Havlin, S and D Ben-Avraham (June 1982). “New approach to self-avoiding walks as a critical phenomenon”. In: *Journal of Physics A: Mathematical and General* 15.6, pp. L321–L328. DOI: 10.1088/0305-4470/15/6/013. URL: https://doi.org/10.1088/0305-4470/15/6/013.

— (Sept. 1982). “Theoretical and numerical study of fractal dimensionality in self-avoiding walks”. In: *Phys. Rev. A* 26 (3), pp. 1728–1734. DOI: 10.1103/PhysRevA.26.1728. URL: https://link.aps.org/doi/10.1103/PhysRevA.26.1728.

de Carvalho, C.Aragao, S. Caracciolo, and J. Frohlich (1983). “Polymers and $g|\phi|^4$ theory in four dimensions”. In: *Nuclear Physics B* 215.2, pp. 209–248. ISSN: 0550-3213. DOI: https://doi.org/10.1016/0550-3213(83)90213-4. URL: https://www.sciencedirect.com/science/article/pii/0550321383902134.

Berretti, Alberto and Alan D. Sokal (Aug. 1985). “New Monte Carlo method for the self-avoiding walk”. In: *Journal of Statistical Physics* 40.3, pp. 483–531. ISSN: 1572-9613. DOI: 10.1007/BF01017183. URL: https://doi.org/10.1007/BF01017183.

Aizenman, Michael (1986). “Rigorous studies of critical behavior”. In: *Physica A: Statistical Mechanics and its Applications* 140.1, pp. 225–231. ISSN: 0378-4371. DOI: https://doi.org/10.1016/0378-4371(86)90226-8. URL: https://www.sciencedirect.com/science/article/pii/0378437186902268.

Duplantier, Bertrand (Aug. 1986). “Polymer Network of fixed topology: renormalization, exact critical exponent γ in two dimensions, and $d = 4 - \epsilon$”. In: *Phys. Rev. Lett.* 57 (8), pp. 941–944. DOI: 10.1103/PhysRevLett.57.941. URL: https://link.aps.org/doi/10.1103/PhysRevLett.57.941.

Madras, Neal and Alan D. Sokal (Jan. 1988). “The pivot algorithm: A highly efficient Monte Carlo method for the self-avoiding walk”. In: *Journal of Statistical Physics* 50.1, pp. 109–186. ISSN: 1572-9613. DOI: 10.1007/BF01022990. URL: https://doi.org/10.1007/BF01022990.

Lee, Sang Bub, Hisao Nakanishi, and Y. Kim (May 1989). “Monte Carlo study of self-avoiding walks on a critical percolation cluster”. In: *Phys. Rev. B* 39 (13), pp. 9561–9572. DOI: 10.1103/PhysRevB.39.9561. URL: https://link.aps.org/doi/10.1103/PhysRevB.39.9561.

Rensburg, E J Janse van and S G Whittington (Dec. 1991). "The BFACF algorithm and knotted polygons". In: *Journal of Physics A: Mathematical and General* 24.23, pp. 5553–5567. DOI: 10.1088/0305-4470/24/23/021. URL: https://doi.org/10.1088/0305-4470/24/23/021.

Diaconis, P., S. Holmes, and R. M. Neal (1997). "Analysis of a non-reversible Markov chain sampler". In: *Technical Report BU-1385-M*.

Sokal, A. (1997). "Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms". English. In: *Functional Integration*. Ed. by Cecile DeWitt-Morette, Pierre Cartier, and Antoine Folacci. Vol. 361. NATO ASI Series. New York, NY, USA: Springer US, pp. 131–192. ISBN: 978-1-4899-0321-1. DOI: 10.1007/978-1-4899-0319-8_6. URL: http://dx.doi.org/10.1007/978-1-4899-0319-8_6.

Chen, Fang, Laszlo Lovasz, and Igor Pak (May 1999). "Lifting Markov Chains to Speed up Mixing". In: *Proceedings of the ACM symposium on Theory of Computing*, pp. 275–281.

Newman, M. E. J. and Gerard T. Barkema (1999). *Monte Carlo Methods in Statistical Mechanics*. Clarendon Press.

Adamic, Lada A. et al. (Sept. 2001). "Search in power-law networks". In: *Phys. Rev. E* 64 (4), p. 046135. DOI: 10.1103/PhysRevE.64.046135. URL: https://link.aps.org/doi/10.1103/PhysRevE.64.046135.

Owczarek, A L and T Prellberg (July 2001). "Scaling of self-avoiding walks in high dimensions". In: *Journal of Physics A: Mathematical and General* 34.29, pp. 5773–5780. DOI: 10.1088/0305-4470/34/29/303. URL: https://doi.org/10.1088/0305-4470/34/29/303.

Lawler, Gregory F., Oded Schramm, and Wendelin Werner (Apr. 2002). "On the scaling limit of planar self-avoiding walk". In: *arXiv Mathematics e-prints*, math/0204277, math/0204277. arXiv: math/0204277 [math.PR].

Rechnitzer, A and E J Janse van Rensburg (Oct. 2002). "Canonical Monte Carlo determination of the connective constant of self-avoiding walks". In: *Journal of Physics A: Mathematical and General* 35.42, pp. L605–L612. DOI: 10.1088/0305-4470/35/42/103. URL: https://doi.org/10.1088/0305-4470/35/42/103.

Hsu, Hsiao-Ping and Peter Grassberger (2004). "Polymers confined between two parallel plane walls". In: *The Journal of Chemical Physics* 120.4, pp. 2034–2041. DOI: 10.1063/1.1636454. eprint: https://doi.org/10.1063/1.1636454. URL: https://doi.org/10.1063/1.1636454.

Rechnitzer, A and E J Janse van Rensburg (Oct. 2008). "Generalized atmospheric Rosenbluth methods (GARM)". In: *Journal of Physics A: Mathematical and Theoretical* 41.44, p. 442002. DOI: 10.1088/1751-8113/41/44/442002. URL: https://doi.org/10.1088/1751-8113/41/44/442002.

Rensburg, E J Janse van and A Rechnitzer (Feb. 2008). "Atmospheres of polygons and knotted polygons". In: *Journal of Physics A: Mathematical and Theoretical* 41.10, p. 105002. DOI: 10.1088/1751-8113/41/10/105002. URL: https://doi.org/10.1088/1751-8113/41/10/105002.

Levin, David A., Yuval Peres, and Elizabeth L. Wilmer (2009). *Markov Chains and Mixing Times*. Providence, RI, USA: American Mathematical Society.

Rensburg, E J Janse van (July 2009). "Monte Carlo methods for the self-avoiding walk". In: *Journal of Physics A: Mathematical and Theoretical* 42.32, p. 323001. DOI: 10.1088/1751-8113/42/32/323001. URL: https://doi.org/10.1088/1751-8113/42/32/323001.

Rensburg, E J Janse van and A Rechnitzer (July 2009). “Generalized atmospheric sampling of self-avoiding walks”. In: *Journal of Physics A: Mathematical and Theoretical* 42.33, p. 335001. DOI: 10.1088/1751-8113/42/33/335001. URL: https://doi.org/10.1088/1751-8113/42/33/335001.

Goodman, Jonathan and Jonathan Weare (Jan. 2010). “Ensemble samplers with affine invariance”. In: *Communications in Applied Mathematics and Computational Science* 5.1, pp. 65–80. DOI: 10.2140/camcos.2010.5.65.

Turitsyn, K. S., M. Chertkov, and M. Vucelja (2011). “Irreversible Monte Carlo algorithms for efficient sampling”. In: *Physica D Nonlinear Phenomena* 240, pp. 410–414.

Sakai, Y. and K. Hukushima (June 2013). “Dynamics of One-Dimensional Ising Model without Detailed Balance Condition”. In: *Journal of the Physical Society of Japan* 82.6, pp. 064003-1–8. DOI: 10.7566/JPSJ.82.064003. arXiv: 1301.4801 [cond-mat.stat-mech].

Thompson, Colin J. (2015). *Mathematical Statistical Mechanics*. Princeton University Press. ISBN: 9781400868681. DOI: doi:10.1515/9781400868681. URL: https://doi.org/10.1515/9781400868681.

Bierkens, J. (2016). “Non-reversible Metropolis-Hastings”. In: *Statistics and Computing* 26.6, pp. 1213–1228. ISSN: 1573-1375. DOI: 10.1007/s11222-015-9598-x. URL: http://dx.doi.org/10.1007/s11222-015-9598-x.

Hu, Hao, Xiaosong Chen, and Youjin Deng (Dec. 2016). “Irreversible Markov chain Monte Carlo algorithm for self-avoiding walk”. In: *Frontiers of Physics* 12.1, p. 120503. ISSN: 2095-0470. DOI: 10.1007/s11467-016-0646-6. URL: https://doi.org/10.1007/s11467-016-0646-6.

Vucelja, M. (2016). “Lifting–A nonreversible Markov chain Monte Carlo algorithm”. In: *American Journal of Physics* 84.12, pp. 958–968. DOI: http://dx.doi.org/10.

1119/1.4961596. URL: http://scitation.aip.org/content/aapt/journal/ajp/84/12/10.1119/1.4961596.

Bierkens, Joris and Gareth Roberts (Apr. 2017). "A piecewise deterministic scaling limit of lifted Metropolis–Hastings in the Curie–Weiss model". In: *Ann. Appl. Probab.* 27.2, pp. 846–882. DOI: 10.1214/16-AAP1217. URL: https://doi.org/10.1214/16-AAP1217.

Kapfer, Sebastian C. and Werner Krauth (Dec. 2017). "Irreversible Local Markov Chains with Rapid Convergence towards Equilibrium". In: *Phys. Rev. Lett.* 119 (24), p. 240603. DOI: 10.1103/PhysRevLett.119.240603. URL: https://link.aps.org/doi/10.1103/PhysRevLett.119.240603.

Guyon, I. et al., eds. (2017). *A-NICE-MC: Adversarial Training for MCMC*. Curran Associates, Inc., pp. 5140–5150. URL: http://papers.nips.cc/paper/7099-a-nice-mc-adversarial-training-for-mcmc.pdf.

de Guzzi Bagnato, Guilherme, José Ricardo Furlan Ronqui, and Gonzalo Travieso (2018). "Community detection in networks using self-avoiding random walks". In: *Physica A: Statistical Mechanics and its Applications* 505, pp. 1046–1055. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2018.04.006. URL: https://www.sciencedirect.com/science/article/pii/S0378437118304242.

Zhou, Zongzheng, Jens Grimm, Sheng Fang, et al. (Oct. 2018). "Random-Length Random Walks and Finite-Size Scaling in High Dimensions". In: *Phys. Rev. Lett.* 121 (18), p. 185701. DOI: 10.1103/PhysRevLett.121.185701. URL: https://link.aps.org/doi/10.1103/PhysRevLett.121.185701.

Campbell, S and E J Janse van Rensburg (June 2020). "Parallel PERM". In: *Journal of Physics A: Mathematical and Theoretical* 53.26, p. 265005. DOI: 10.1088/1751-8121/ab8ff7. URL: https://doi.org/10.1088/1751-8121/ab8ff7.

Zhou, Zongzheng, Jens Grimm, Youjin Deng, et al. (2020). *Random-length Random Walks and Finite-size Scaling on high-dimensional hypercubic lattices I: Periodic Boundary Conditions*. arXiv: 2008.00913 [math-ph].

Fang, Sheng, Youjin Deng, and Zongzheng Zhou (2021). *Logarithmic Finite-Size Scaling of the Self-avoiding Walk at Four Dimensions*. arXiv: 2103.04340 [cond-mat.stat-mech].

Appendices

Appendix A

Performance in 4D and 5D

We investigated the performance of the *three-move nonreversible Berretti-Sokal algorithm* in $4D$ and $5D$. We did not find it to be efficient, when compared to the *two-move nonreversible Berretti-Sokal algorithm.* The detailed findings are in the table.

dimension $d = 4$		
system size n	$x = 6/35$	$x = 3/35$
25	0.714 ± 0.069	2.970 ± 0.356
51	1.081 ± 0.050	2.216 ± 0.229
75	0.994 ± 0.033	2.812 ± 0.658
101	0.945 ± 0.028	2.349 ± 0.190
dimension $d = 5$		
system size n	$x = 1/5$	$x = 1/10$
21	0.920 ± 0.002	4.214 ± 1.108
25	0.961 ± 0.001	4.451 ± 0.571
31	0.992 ± 0.002	4.992 ± 0.696
35	0.995 ± 0.002	3.261 ± 0.513

Table A.1: The ratio of integrated autocorrelation times of the *three-move nonreversible Berretti-Sokal algorithm*, τ, and the *two-move nonreversible Berretti-Sokal algorithm*, τ_0, for $4D$ and $5D$ systems as a function of the linear system size n, the SAW unit length weight x. The ratio about 1 for ($x = 6/35$, $d = 4$) and ($x = 1/5$, $d = 5$), however for $(x = 3/35, d = 4)$ and $(x = 1/10, d = 5)$ it is above unity, which indicates that two-mode nonreversible Berretti-Sokal algorithm is more efficient there.