An Analysis of Polymers Modeled as Self-Avoiding Random Walks

Pranav Parakh

Dec 4, 2024

Abstract

Polymers, have diverse applications in everyday life and materials science. To understand their properties, we often model them as self-avoiding random walks (SAWs), capturing the physical constraint that no two monomers can occupy the same space. This study explores the behavior of polymers using SAWs, focusing on critical exponents and scaling behavior. We implemented algorithms to generate and enumerate SAWs in 2, 3, and 4 dimensions, estimating parameters like the connective constant μ, the critical exponent γ, and the Flory exponent ν. Through Monte Carlo simulations and the Clisby method, we investigated the end-to-end distance, radius of gyration, and fractal dimensions of SAWs. Our findings highlight the significant role of dimensionality in polymer behavior, providing insights into their geometric and statistical properties. Specifically, we found $\gamma = 1.451 \pm 0.005$, $\nu = 0.76 \pm 0.06$, and $\mu = 2.574 \pm 0.004$ in 2D; $\gamma = 1.151 \pm 0.006$, $\nu = 0.63 \pm 0.03$, and $\mu = 4.642 \pm 0.005$ in 3D; and $\gamma = 1.092 \pm 0.007$, $\nu = 0.55 \pm 0.04$, and $\mu = 6.658 \pm 0.003$ in 4D. While our results align well with theoretical values, they are not consistently within one standard deviation of the accepted values. This suggests future research directions which includes increasing the simulation precision and length, exploring different lattice geometries, and incorporating different polymer models.

1 Introduction

Polymers are large-chain macromolecules consisting of repeating smaller units known as monomers. Polymers are ubiquitous in everyday life; low density polyethylene (LDPE), for example, makes up much of the soft plastics we use in everyday life. LDPE is a polymer that ranges from hundreds to tens of thousands of repeating C_2H_4 units.

When studying polymers theoretically, it is interesting to consider the large length limit, where we consider the asymptotic behavior of the polymer as the number of monomers (N) increases. Because physical polymers live in 3 dimensional space, they naturally cannot overlap with each other in the same physical location. This

leads us to the use of self-avoiding walks (SAWs) as a model for polymers. The self-avoiding condition ensures that a walk will never retrace its steps, thus encoding the condition that two physical molecules cannot be in the same place at once. Although the chemical properties of polymers are highly dependent on their constituent monomer units, their physical behaviour is largely geometrically determined. Thus, by studying the behaviour of SAWs, we aim to gain intuition about the physical properties of polymers.

One of the most natural questions that arises is how to count the number of SAWs for a given number of monomers N. In this work, we will repeatedly draw analogies to simple random walks to gain familiarity with the properties we are considering, and then we will use those to draw conclusions about SAWs. The number of random walks on a D dimensional lattice is easy to enumerate. At each step, the walk can "choose" to go in $2D$ directions [11]. Thus, for a random walk of length N, it is trivial to see that the number of random walks is $2D^N$. It is unclear how to do such a simple enumeration for self-avoiding walks; in fact, there is no known closed-form expression for enumerating them as we can with random walks.

The number of SAWs of length N, c_N, has been shown to grow exponentially in N in the form

$$c_N = A\mu^N N^{\gamma-1} \tag{1}$$

for a simple cubic lattice in D dimensions, where μ is the connective constant, A is some amplitude, and γ is a universal critical exponent. A and μ are lattice dependent, while γ is believed to be universal for all lattices in a given dimension D [4].

We might also care about the "size" of a polymer of length N, which can be measured by the average distance between the two ends of the polymer. For a simple random walk, it is clear that the end-to-end distance scales as $\sqrt{N}$. It is predicted that this quantity scales with a power law for SAWs, and we will attempt to calculate the relavant critical exponent. Some other interesting parameters that we may wish to examine are the radius of gyration of a polymer, which can be thought to measure how "curled up" a polymer is, and the average distance of each monomer in the chain from the endpoints. All three of these measures are expected in SAWs to scale with the same critical exponent, ν, known as the Flory exponent [6].

There have been several rigorous estimates of μ, γ, and ν in 2 and 3 dimensions. ν and γ are critical exponents that are believed to be universal in that they only depend on the dimension of the underlying lattice, while μ depends on the geometry of the lattice. In 2D, γ and ν are believed to be $\frac{43}{32}$ and $\frac{3}{4}$ respectively [6], and for a hypercubic lattice, μ has been estimated at 2.638 [4]. In 3D, γ and ν have been computed to be 1.1575 and 0.587579 [2, 3], and μ has been estimated at 4.684 [4]. In this work, we aim to reproduce these results in 2 and 3 dimensions, and compute these values in 4 dimensions as well.

2 Theoretical Background

We now present a rigorous mathematical model for the self-avoiding walk and explain in detail how the above mentioned critical exponents are calculated.

An N-step SAW on a hypercubic lattice of dimension D, $\mathbb{Z}^d$ can be represented as a function from $\omega : \{0, ..., N\} \to \mathbb{Z}^d$ with $|w(j+1) - w(j)| = 1$ for all $1 \leq j \leq N$ and $w(i) \neq w(j)$ for all $i \neq j$ [6].

2.1 Connective Constant

As discussed previously, the the number of SAWs for a given length N, c_N follows the exponential form shown in equation 1. In principle, it should be simple to confirm this and find values for μ and γ – we just need to enumerate the number of SAWs at any given N. This turns out to be a difficult problem. We present two approaches to it in section 3.

2.2 Flory Exponent

The mean squared end-to-end distance, $\langle R^2_{rms} \rangle_N$, can be represented by

$$\begin{aligned} \langle R^2_{rms} \rangle_N &= \frac{1}{c_N} \sum_{\Omega_N} (\omega_0 - \omega_n)^2 \\ &= CN^{2\nu} \end{aligned} \tag{2}$$

where c_N is the connective constant that enumerate the number of SAWs of length N, Ω_N denotes a sum over all SAWs of length N, C is a constant, and ν is the Flory exponent [3, 6].

The mean-squared radius of gyration, $\langle R^2_g \rangle_N$ can be written as

$$\begin{aligned} \langle R^2_g \rangle_N &= \frac{1}{c_N} \sum_{\Omega_N} \left[\frac{1}{N} \sum_{i=1}^{N} (\mathbf{r}_i - \mathbf{r}_{\mathrm{cm}})^2 \right] \\ &= DN^{2\nu} \end{aligned} \tag{3}$$

where N is the number of steps in the walk, $\mathbf{r}_i$ is the position vector of the ith step, $\mathbf{r}_{\mathrm{cm}}$ is the position vector of the center of mass of the walk, D is some constant. $\mathbf{r}_{cm}$ is given by

$$\mathbf{r}_{cm} = \sum_{i}^{N} \frac{\mathbf{r}_i}{N}. \tag{4}$$

The radius of gyration R_g gives an indication of how spread out the points of the walk are from the center of mass.

Finally, we can write the expression for the mean-squared distance of a monomer from the endpoints of a polymer, $\langle R_m^2 \rangle_N$, as

$$\begin{aligned} \langle R_m^2 \rangle_N &= \frac{1}{c_N} \sum_{\Omega_N} \left[\frac{1}{2(N+1)} \sum_{j=0}^{n} \left[(\omega_0 - \omega_j)^2 + (\omega_N - \omega_j)^2 \right] \right] \\ &= F N^{2\nu} \end{aligned} \tag{5}$$

where N is the number of steps in the walk, ω_j is the position of the jth monomer, ω_0 is the position of the starting point of the walk, ω_N is the position of the end point of the walk, and F is some constant[6].

3 Simulation methods

To estimate the critical behaviour of SAWs, it is necessary to be able to both enumerate them and generate them in various dimensions. In this section, we discuss algorithms to do both.

3.1 Exact Enumeration

There is a very naive algorithm that we implement in 2, 3, and 4 dimensions in order to enumerate exactly the number of SAWs. It goes as follows:

1. Choose an initial position
2. Take a step in each of the $2D$ directions possible, where D is the dimension of the hypercubic lattice
3. Check if each of the new step satisfies the self-avoidance condition. If not, terminate that walk.
4. If the length of the new walk is the desired length, count it and terminate that walk
5. If not, repeat steps 2 through 4 from each of the $2D$ new positions

Clearly, this algorithm is not very efficient, as the number of recursive calls scales as $2D^N$ and has exponential time and memory complexity. Regardless, we can run this brute force algorithm for 2, 3 and 4 dimensions.

3.2 Pivot Monte-Carlo for Generating Walks

To accurately estimate the critical behaviour, we want to be able to sample SAWs *uniformly* from the space of all SAWs for a given dimension D and length N. One might initially think that we can sample uniformly using the method from the previous algorithm, where we choose an initial point, and at random, choose one of the $2D$ directions to move to. Unfortunately, due to the self-avoidance condition, this algorithm does not uniformly sample from the entire space. This is most easly seen

from an example given in ref. [11]: a walk in 2D where we go North, then East 3 times has a probability of $\frac{1}{4}\times\frac{1}{3}\times\frac{1}{3}\times\frac{1}{3}=\frac{1}{108}$, whereas a walk that goes NESS (where the letters stand for the cardinal directions) has probability $\frac{1}{4}\times\frac{1}{3}\times\frac{1}{3}\times\frac{1}{2}=\frac{1}{72}$. Since these two walks don't have the same probability, we are clearly not taking a uniform sample over the state space.

An alternate algorithm that is ergodic is the pivot algorithm, invented by Lal in 1969, and used to simulate SAWs by Madras and Sokal. [8, 9]. The pivot algorithm implements the following steps for simulating a SAW of length N on a hypercubic lattice in D dimensions:

1. Begin with any configuration of length N. In this simulation, we use a rod in D dimensions.
2. Choose a random pivot point i, $1 < i < N$ with uniform probability
3. Choose with uniform probability which side of the pivot we will apply the rotation to
4. Choose any of the symmetry operations in D dimensions at random and apply it to the side of the polymer that was selected. Enumeration of the symmetry operations in 2, 3, and 4, dimensions is given in Appendix A
5. Check if the new generated walk is a SAW. If not, discard the operation, and repeat from step 2
6. If the walk is valid, repeat from step 2 until the desired number of pivots has been reached

Applying this algorithm will generate a uniform sample of SAWs from the state space, and is the algorithm we implement in this work.

3.3 Clisby Method for estimating c_N

The exact enumeration algorithm becomes computationally intractable for any large N, and so, to see the critical exponent behaviour, we must use an estimation algorithm to find c_N for larger values of N. Clisby provides such an algorithm in Ref. [4], which is outlined as follows:

1. Generate two SAWs of length N, ω_1 and ω_2
2. Place the endpoint of one of the walks at the origin, and the start of the other walk 1 unit of distance away from the origin. Denote this operation by $\omega_1 \circ \omega_2$.
3. Calculate an indicator function $B(\omega_1, \omega_2)$ as follows:

$$B_N(\omega_1,\omega_2)=\begin{cases}0 & \text{if } \omega_1\circ\omega_2 \text{ not self-avoiding}\\ 1 & \text{if } \omega_1\circ\omega_2 \text{ self-avoiding}\end{cases} \tag{6}$$

4. Repeat steps 1 through 3 for many walks and find the average value of B_N

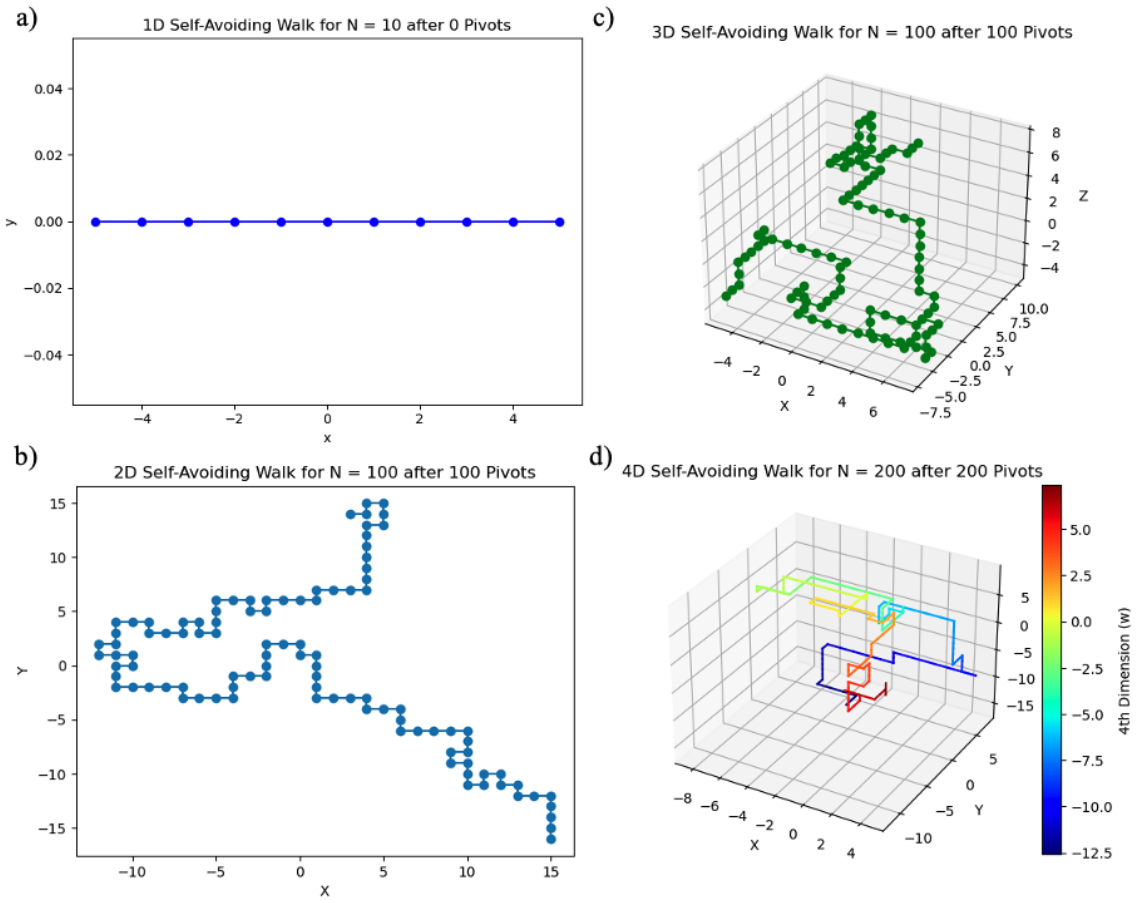

Figure 1: Example SAWs in various dimensions. Figures (a), (b), and (c) show SAWs in 1, 2, and 3 dimensions. Figure (d) shows a 3 dimensional projection of a 4 dimensional walk, where the 4th dimension is shown on the colour axis.

Following the derivation provided in Ref. [4], we define

$$B_{M,N} \equiv \text{Mean value of } B(\omega_1, \omega_2) \text{ over all pairs of } M \text{ and } N \text{ step walks}, \quad (7)$$

and further write

$$\tilde{B}_N \equiv \Omega B_N = \frac{c_{2N+1}}{c_N^2}, \quad (8)$$

where Ω is the coordination number of the lattice. It is now clear to see that we can estimate c_{2N+1} if we know c_N and can compute B_N

4 Computational Results

In this section, we implement all of the algorithms discussed in the previous section. All simulation code was written in Python and is available in Appendix B.

4.1 Generating SAWs

First, we implement the Pivot Monte-Carlo algorithm in order to generate SAWs in various dimensions. In $D = 1$, the self avoiding walks are trivial, as the only possible walk is a straight line. We plot example SAWs for $D = 1, 2, 3$ and 4 in Figure 1. By choosing a number of pivots almost equal to the length of the monomer, we ensure that we have a uniformly random polymer generated at each step. The 4 dimensional case is visualized by using colour as the fourth dimension.

4.2 Connective Constant

In this work, we first enumerate the number of SAWs exactly using the algorithm in section 3.1 and then extend to larger N using the Clisby method described in section 3.3.

4.2.1 Exact Enumeration

When enumerating SAWs exactly in 2, 3, and 4 dimensions, we ran the program on a high-performance computing cluster for 48 hours distributed over many nodes. This allowed us to enumerate the number of SAWs exactly up to N = 24, 15, and 13 for dimensions of 2, 3, and 4 respectively. These results are shown in Figure 2 (a), (b), and (c), and also provided exactly in Table 1. We fit Equation 1 to these values and obtain values for μ and gamma. These values are shown in Table 2, in the columns labeled "Exact".

The exact enumeration of SAWs are shown Table 1, which match exactly with Reference [10] and other references.

4.2.2 Clisby Method

Since we could only compute the exact number of SAWs for a relatively small N, we use the method described in section 3.3 to estimate c_N for much larger N. In 2 dimensions, we were only able to compute up to $N = 24$. We can then use Equation 8 to write following:

$$c_{49} = \tilde{B}_{24} \cdot c_{24}^2 \tag{9}$$

$$c_{99} = \tilde{B}_{49} \cdot c_{49}^2 \tag{10}$$

$$c_{199} = \tilde{B}_{99} \cdot c_{99}^2 \tag{11}$$

$$c_{399} = \tilde{B}_{199} \cdot c_{199}^2, \tag{12}$$

where we use Pivot Monte Carlo to estimate that $\tilde{B}_{24} = 1.85 \pm 0.04$, $\tilde{B}_{49} = 1.59 \pm 0.03$, $\tilde{B}_{99} = 1.46 \pm 0.04$, $\tilde{B}_{199} = 1.25 \pm 0.04$.

We do the same for three and four dimensions, and find that in 3D, $\tilde{B}_{15} = 2.826 \pm 0.05$, $\tilde{B}_{31} = 3.084 \pm 0.03$, $\tilde{B}_{63} = 2.304 \pm 0.04$, and $\tilde{B}_{127} = 2.712 \pm 0.04$, and in 4D, $\tilde{B}_{13} = 5.392 \pm 0.05$, $\tilde{B}_{27} = 5.096 \pm 0.04$, $\tilde{B}_{55} = 5.472 \pm 0.04$, and $\tilde{B}_{111} = 5.488 \pm 0.04$.

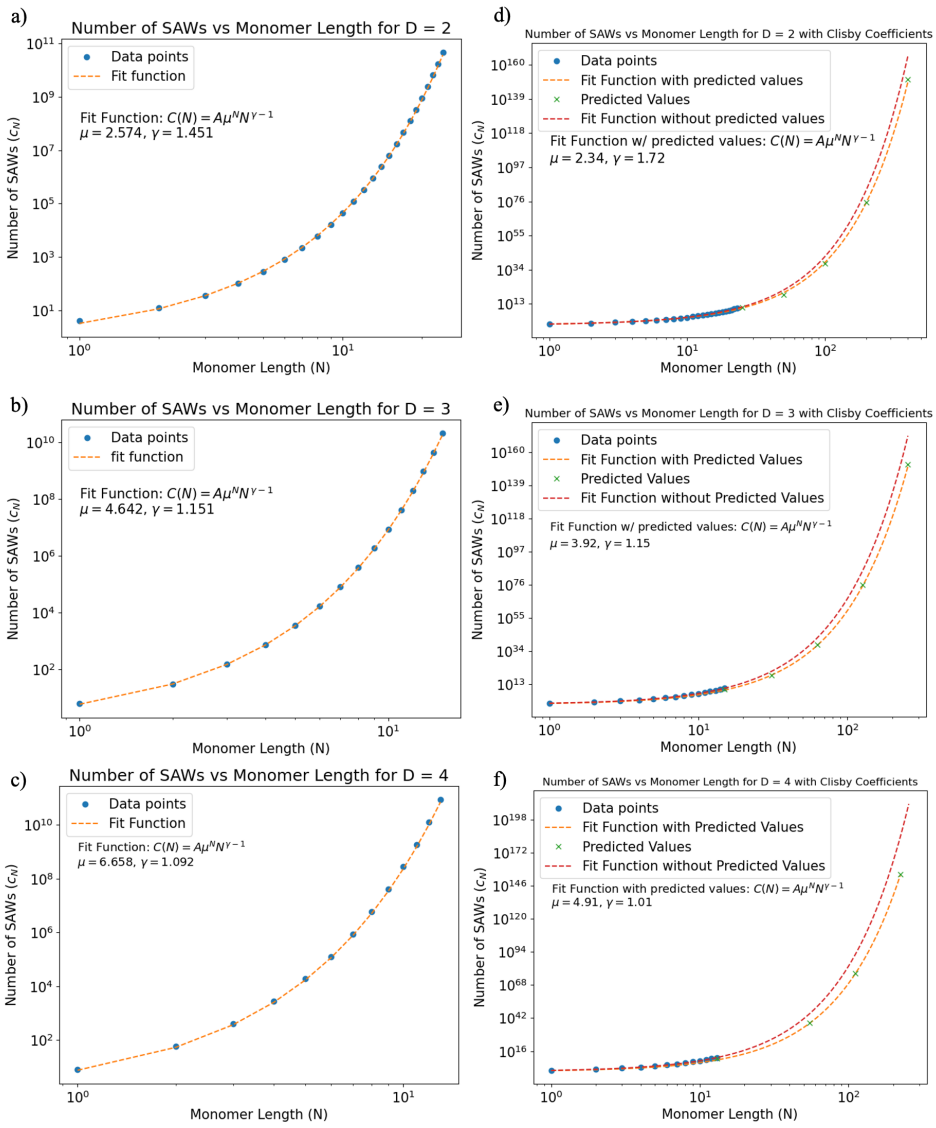

Figure 2: Plots (a), (b), and (c) show the c_N versus N computed exactly, fit to Equation 1 in varying dimension. (d), (e), and (f) are extensions of the previous plots that use the Clisby method to generate c_N for higher N, fit to the same equation.

Using these values and Equation 8, we estimate larger c_N values for 2, 3, and 4 dimensions and plot them in Figure 2 (d), (e), and (f). In orange in these figures we fit the new data (including the predicted c_N values) to Equation 1 and write the values obtained for the critical parameters in Table 2.

There is some discrepancy between the critical values when we include the higher N counts versus when we do not. There could be several reasons for this; due to computational complexity, we did not run the Monte Carlo simulation for B_N for very many iterations, and thus likely have large error in those points. It is also probable that we did not extend our analysis to high enough N for the asymptotic

Table 1: Number of SAWs in 2, 3, and 4 Dimensions

N	**2D**	**3D**	**4D**
1	4	6	8
2	12	30	56
3	36	150	392
4	100	726	2696
5	284	3534	18584
6	780	16926	127160
7	2172	81390	871256
8	5916	387966	5946200
9	16268	1853886	40613816
10	44100	8809878	276750536
11	120292	41934150	1886784200
12	324932	198842742	12843449288
13	881500	943974510	87456597656
14	2374444	4468911678	-
15	6416596	21175146054	-
16	17245332	-	-
17	46466676	-	-
18	124658732	-	-
19	335116620	-	-
20	897697164	-	-
21	2408806028	-	-
22	6444560484	-	-
23	17266613812	-	-
24	46146397316	-	-

Table 2: Critical Exponents for Dimensions $D = 2, 3$, and 4

Dimension (D)	**μ Exact**	**γ Exact**	**μ Predicted**	**γ Predicted**
2	2.574 ± 0.004	1.451 ± 0.005	2.34 ± 0.02	1.72 ± 0.01
3	4.642 ± 0.005	1.151 ± 0.006	3.92 ± 0.05	1.15 ± 0.02
4	6.658 ± 0.003	1.092 ± 0.007	4.91 ± 0.07	1.01 ± 0.03

behaviour to be truly accurate. While we recover the expected trends, we do not to within one standard deviation recover the accepted values for μ and γ.

4.3 Flory Exponent

We use the method described in section 3.2 to generate many SAWs of varying length N. For each value of N we choose, we average the end-to-end distance, the radius of gyration, and the distance from the each monomer to the endpoints over 400 iterations. We then calculate $\langle R^2_{rms} \rangle$, $\langle R^2_g \rangle$, and $\langle R^2_m \rangle$ and plot them in Figure 3 (a), (b), and (c). We see that all of the properties seem to grow exponentially as

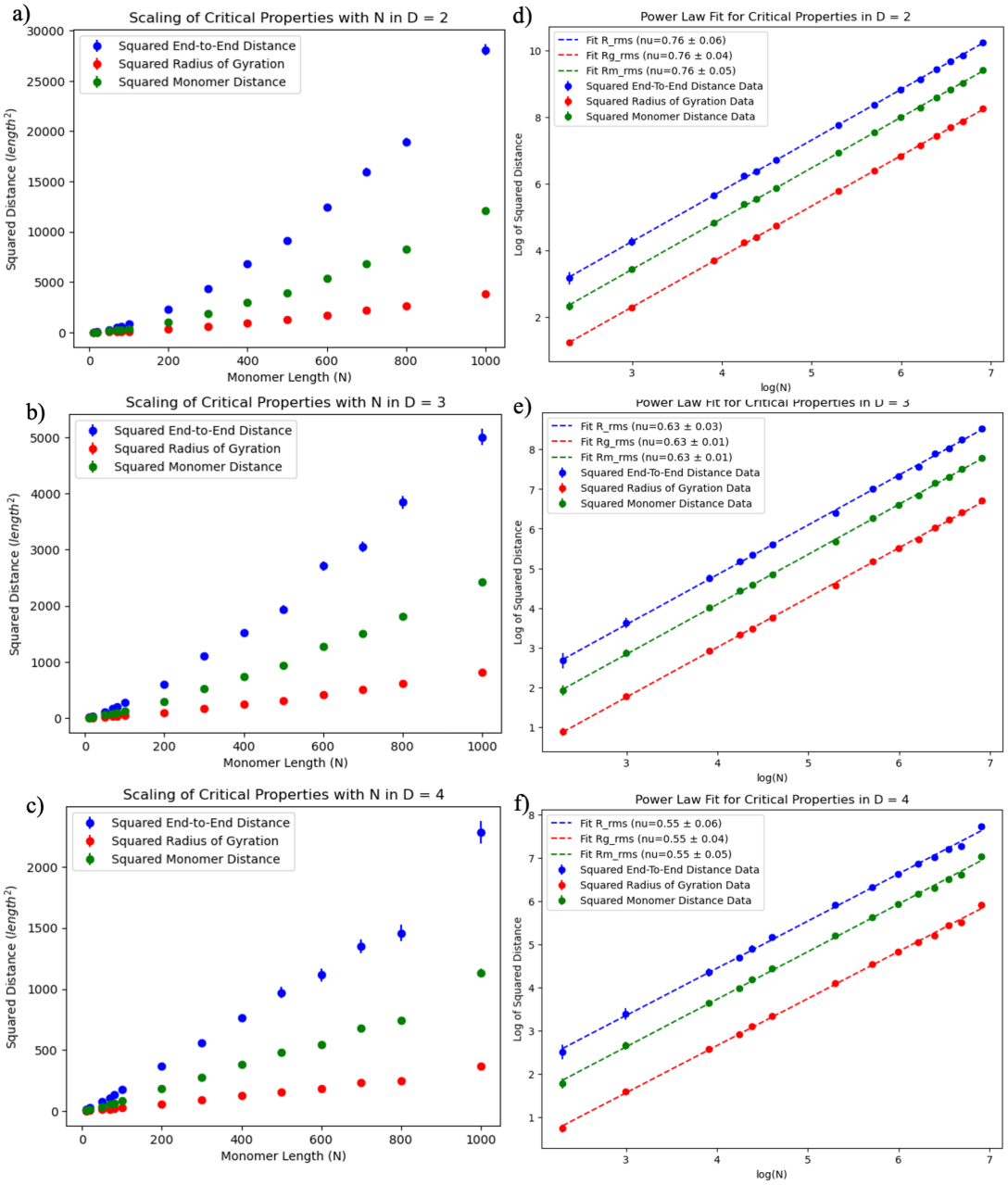

Figure 3: Plots (a), (b), and (c) show three different critical properties of SAWs versus N in 2, 3, and 4 dimensions respectively. (d), (e), and (f) make the same plots in log scale, and fit the data with Equations 2, 3, and 5 and show the value for the critical exponent ν.

expected, but at different rates. When we plot them in log scale as is done in Figure 3 (d), (e), and (f), we see that we can fit them all with a line that has approximately the same slope. From Equations 2, 3, and 5, we know that the slope of this fit is going to be proportional to twice the critical exponent ν, and thus we can extract values of ν in 2, 3, and 4 dimensions by taking the average of these three values. These values are reported in Table 3.

Table 3: Flory Exponent ν versus Literature value

Dimension (D)	**ν calculated**	**ν accepted**
2	0.76 ± 0.06	0.75
3	0.63 ± 0.03	0.588
4	0.55 ± 0.04	0.5

In 2 dimensions, we recover the expected exponent to within one standard deviation, but for higher dimensions, we do not. This could be again because we did not consider large enough chains for the asymptotic behaviour of the critical exponents to dominate over small-scale fluctuations.

5 Discussion

It is interesting to note that while in 2 and 3 dimensions, the Flory exponent is different from the analog in random walks, in 4 dimensions and higher they become equal. This is because in 4 dimensions, since there are $2 \times 4 = 8$ directions in which a walk can grown in, it is increasingly unlikely for even a random walk to intersect itself. Thus, we see that the self-avoidance condition, which is strong in 2 and 3 dimensions, has a diminishing effect in 4 dimensions which renders the large N behaviour identical to random walks. We expect this would continue to be the case in $D \geq 4$.

Self-avoiding walks, in the limit where the length of the chain becomes very large, exhibit some fractal tenancies. This is because the radius of gyration scales with an exponent that is less than linear in N, so the polymer essentially becomes more "curled up" as you add more monomers. The fractal dimensions quantifies the space-filling properties of the SAW. To be more clear, the fractal dimension of a polymer can be visualized as the exponent describing the exponential increase in mass of a polymer enclosed in a sphere of radius R as the radius increases [7]. As we discussed in class with the Ising model at its critical temperature, there is a scale invariance for the pockets of order in this system. The fractal dimensions aims to quantify this scale invariance [12]. We can visualize this in Figure 4, where we have plotted a very large SAW of length 10^4 in plot (a), and then show a zoomed in version of the same polymer in plot (b). As we can see, as we zoom in, the polymer properties remain invariant, giving a visual description of the scale invariance that the fractal dimension is trying to quantify.

For a more rigorous description of the fractal dimension, we turn the reader towards

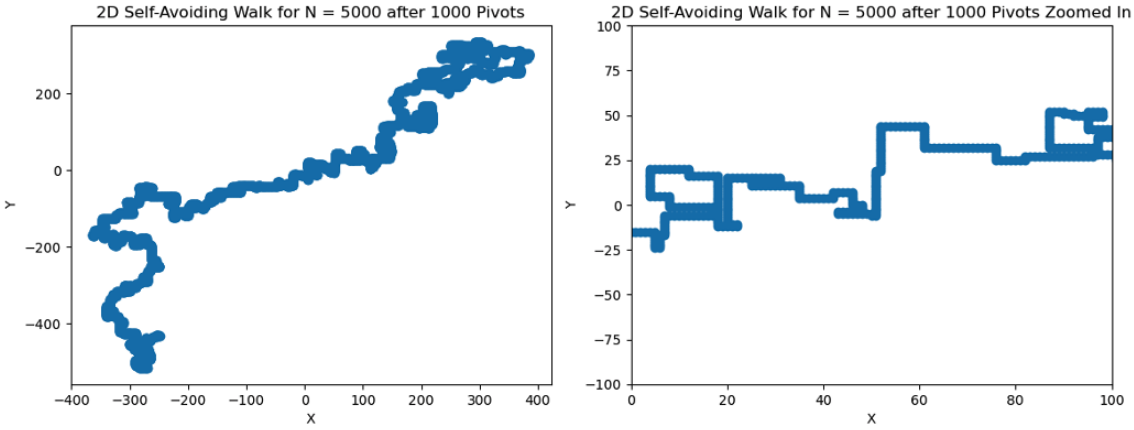

Figure 4: (a) shows a 2D SAW with $N = 5000$. (b) shows a zoomed in version of the same polymer, intended to show the scale invariance properties of the SAW.

Reference [5]. To gain some intuition about how the fractal dimension of a relates to critical exponents, we will once again consider random walks. We will denote the fractal dimension by D_f. For a line, $D_f = 1$. For a plane, $D_f = 2$, and for a solid volume, $D_f = 3$. These clearly agree with our intuitive idea of dimensionality. But now we can ask about the dimension of a random walk in 2 dimensions, where the length of the walk is taken to be asymptotically large. Recall that for a random walk, the radius of gyration $R_g \propto \sqrt{N}$. This means that if we were to zoom into any particular section of a random walk with very large N, it would continue to look geometrically the same. Since $R_g^2 \propto N^{2\nu}$ where ν is the Flory exponent, the Flory exponent of a random walk is $\nu = \frac{1}{2}$. The relationship between the Flory exponent and the fractal dimension of a polymer is given by

$$D_f = \frac{1}{\nu}, \tag{13}$$

and so we see that the fractal dimension of a random walk is 2 [1, 7]. Thus, even though a random walk is a 1 dimensional line at finite N, its asymptotic behaviour tends to fill a plane.

Let's ask the same question now about SAWs. We might expect that because of the self avoidance condition, SAWs are "less" space-filling than random walks. This intuition would be correct. In fact, in 2 dimensions, using Equation 13, we see that

$$D_f^{2d} = \frac{1}{0.76} = 1.32 \pm 0.08.$$

In 3 and 4 dimensions, we see that

$$D_f^{3d} = \frac{1}{0.63} = 1.59 \pm 0.09.$$

$$D_f^{4d} = \frac{1}{0.55} = 1.8 \pm 0.1.$$

This matches with our expectations, and with the heuristic argument we gave for convergence in $D \geq 4$ of SAWs to random walks. In fact, the accepted value for ν in four dimensions is 0.5, which is identical to that of the random walk. We see that in 2 dimensions, the self-avoidance condition is important, and reduces the fractal dimension from a random walk of 2 to 1.32 ± 0.08. This means that although a SAW is more space-filling than a line, it does not fill the entire two dimensional plane. It says that the mass of a SAW in 2 dimensions as enclosed by a sphere of radius R is given by

$$m \propto R^{D_f}.$$

In three dimensions, the fractal dimension is 1.59 ± 0.09, and in 4 dimensions, we find it to be 1.8 ± 0.1. We can see that as we increase dimension, the importance of the self-avoidance condition diminishes, demonstrated by the convergence of the fractal dimension to 2.

6 Conclusion

In this work, we have investigated the behavior and properties of polymers modeled as self-avoiding random walks (SAWs). By distinguishing SAWs from simple random walks through the self-avoidance condition, we've explored various critical exponents and their significance in understanding polymer behavior.

We implemented several algorithms to enumerate and simulate SAWs in 2, 3, and 4 dimensions. The exact enumeration algorithm, although computationally expensive, allowed us to determine the number of SAWs for smaller values of N. We then used the Clisby method to estimate the number of SAWs for larger values of N, enabling us to extract the connective constant μ and the universal critical exponent γ. Our estimations, though not entirely accurate due to computational and other limitations, showed good alignment with expected theoretical values, especially in lower dimensions.

Through Monte Carlo Pivot simulations, we were able to approximate the end-to-end distance, radius of gyration, and distance from endpoints for SAWs. These measurements allowed us to compute the Flory exponent ν, which describes the scaling behavior of SAWs with their length N. Our results align well with theoretical predictions, particularly in 2 dimensions.

Finally, we examined the concept of fractal dimensions, illustrating the space-filling properties and scale invariance of SAWs. These fractal dimensions further emphasize the distinct nature of SAWs compared to random walks, especially in higher dimensions where the effects of self-avoidance diminish.

This study reinforces the understanding that while chemical properties of polymers depend on their monomeric units, their physical properties can be largely understood through geometrical and statistical models like SAWs. Despite some discrepancies, the trends observed in critical exponents and fractal dimensions are in substantial agreement with theoretical predictions.

7 Future Work

There are three major avenues we can use to extend this work. First, we can run the analysis we already did to higher N and with more iterations in an attempt to increase the precision of the properties we extracted. This would likely involve vectorizing and rewriting our code in a more efficient language like C, and using more computation time on a computing cluster.

Next, we can try conducting the same analysis on different lattice shapes. In this work, we only used hypercubic lattices in 2, 3, and 4 dimensions, but it would be interesting to validate the universality of γ on different lattice geometries.

Finally, we can consider alternate polymer models that may more closely match reality. For example, we assumed in this work that polymers are arbitrarily flexible. In reality, we may wish to implement some level of polymer stiffness which limits the types of symmetry operations we can do when pivoting a polymer. This would likely affect the radius of gyration, the Flory exponent, and the fractal dimension.

8 Acknowledgments

ChatGPT helped me comment my code and make pretty plots, and was useful in writing SLURM submit files.

References

[1] Wolfram Demonstrations Project. URL https://demonstrations.wolfram.com/A1DRandomWalkWithFractalDimension20/.

[2] S. Caracciolo, M. S. Causo, and A. Pelissetto. High-precision determination of the critical exponent gamma for self-avoiding walks. *Physical Review E*, 57(2):R1215–R1218, Feb. 1998. ISSN 1063-651X, 1095-3787. doi: 10.1103/PhysRevE.57.R1215. URL http://arxiv.org/abs/cond-mat/9703250. arXiv:cond-mat/9703250.

[3] N. Clisby. Accurate Estimate of the Critical Exponent for Self-Avoiding Walks via a Fast Implementation of the Pivot Algorithm. *Physical Review Letters*, 104(5):055702, Feb. 2010. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.104.055702. URL https://link.aps.org/doi/10.1103/PhysRevLett.104.055702.

[4] N. Clisby. Calculation of the connective constant for self-avoiding walks via the pivot algorithm. *Journal of Physics A: Mathematical and Theoretical*, 46(24):245001, June 2013. ISSN 1751-8113, 1751-8121. doi: 10.1088/1751-8113/46/24/245001. URL http://arxiv.org/abs/1302.2106. arXiv:1302.2106 [cond-mat, physics:math-ph].

[5] P.-G. d. Gennes. *Scaling concepts in polymer physics*. Cornell University Press, Ithaca, 8. print., nachdr edition, 2005. ISBN 9780801412035.

[6] I. Jensen. A new transfer-matrix algorithm for exact enumerations: self-avoiding walks on the square lattice, Sept. 2013. URL http://arxiv.org/abs/1309.6709. arXiv:1309.6709 [cond-mat, physics:math-ph].

[7] D. Johansen, J. Trewhella, and D. P. Goldenberg. Fractal dimension of an intrinsically disordered protein: Small-angle X-ray scattering and computational study of the bacteriophage N protein. *Protein Science : A Publication of the Protein Society*, 20(12):1955–1970, Dec. 2011. ISSN 0961-8368. doi: 10.1002/pro.739. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3302640/.

[8] M. Lal. 'Monte Carlo' computer simulation of chain molecules. I. *Molecular Physics*, 17 (1):57–64, Jan. 1969. ISSN 0026-8976, 1362-3028. doi: 10.1080/00268976900100781. URL http://www.tandfonline.com/doi/abs/10.1080/00268976900100781.

[9] N. Madras and A. D. Sokal. The pivot algorithm: A highly efficient Monte Carlo method for the self-avoiding walk. *Journal of Statistical Physics*, 50(1):109–186, Jan. 1988. ISSN 1572-9613. doi: 10.1007/BF01022990. URL https://doi.org/10.1007/BF01022990.

[10] R. D. Schram, G. T. Barkema, and R. H. Bisseling. Exact enumeration of self-avoiding walks, Apr. 2011. URL http://arxiv.org/abs/1104.2184. arXiv:1104.2184.

[11] G. Slade. Self-Avoiding Walks, Jan. . URL https://personal.math.ubc.ca/~slade/intelligencer.pdf.

[12] D. Stauffer, A. Aharony, and A. Aharonî. *Introduction to percolation theory*. Taylor & Francis, London, rev. 2. ed., transferred to digital print edition, 2003. ISBN 9780748402533.

9 Appendix A: Symmetry Operations

2D Rotation Matrices

In 2D space, rotation matrices for the pivot algorithm involve rotations in the plane:

$$R_{\frac{\pi}{2}} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad R_{\pi} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \quad R_{\frac{3\pi}{2}} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

3D Rotation Matrices

In 3D space, rotation matrices can be defined for rotations around each axis:

$$R_{x,90^\circ} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \quad R_{x,180^\circ} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad R_{x,270^\circ} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix},$$

$$R_{y,90^\circ} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad R_{y,180^\circ} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad R_{y,270^\circ} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix},$$

$$R_{z,90^\circ} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad R_{z,180^\circ} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad R_{z,270^\circ} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

4D Rotation Matrices

In 4D space, rotation matrices are defined for rotations in the planes. Here are the rotation matrices for 90°, 180°, and 270°:

$$R_{xy,90^\circ} = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad R_{xz,90^\circ} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$R_{xw,90^\circ} = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \quad R_{yz,90^\circ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$R_{yw,90^\circ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad R_{zw,90^\circ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$R_{xy,180^\circ} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad R_{xz,180^\circ} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$R_{xw,180^\circ} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \quad R_{yz,180^\circ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$R_{yw,180^\circ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \quad R_{zw,180^\circ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

$$R_{xy,270^\circ} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad R_{xz,270^\circ} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$R_{xw,270^\circ} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}, \quad R_{yz,270^\circ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$R_{yw,270^\circ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}, \quad R_{zw,270^\circ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

10 Appendix B: Code Availability

Code used for generating the data presented in this work is available here: `https://github.com/pdparakh108/SAWs_Critical_Behaviour`