Topics in Computational Polymer Physics

Kamyar (Kam) Modjtahedzadeh
advisor: Professor Alex Klotz

California State University, Long Beach

November 20, 2022

Abstract

Polymer physics studies the mechanical properties and kinetics of monomers and polymers. In this research, we study molecular dynamics and growing self-avoiding walks on basic levels. For the molecular dynamics, we wish to learn how equilibrium poly[n]catenanes behave so we can compare experiments to theory. To do so we simulate [2]catenanes, starting with straightforward simulations and then simulations with more complex collisions. However, before we do that, we make sure that Brownian dynamics algorithms can reproduce known physics to understand the scaling exponents relating physical properties of the polymer systems of interest. When we realize that our Brownian dynamics algorithms work on single polymer chains, we attempt to obtain the equilibrium properties of [2]catenanes and see how hydrodynamics affect them. Moreover, the latter part of this letter; i.e., the growing self-avoiding walk section, is a continuation of a study set about by Wyatt Hooper & Alex Klotz. The growing self-avoiding walk model has only been inspected on a lattice and it is obscure how much of the actual effects are due to the lattice. Therefore, we simulate it off any lattice to find more general principles governing the walk.

Table of Contents

1 Introduction

Materials which are formed by macromolecules are known as polymers. Polymers are an essential part of everyday life (e.g., water bottles and tires). Although there are synthetics polymers like plastic, they also occur naturally such as cellulose [1]. Polymer science is studied theoretically, experimentally in labs, and computationally through simulations. Many scientists focus on computer simulations as this strategy allows for predictions and provides theories of experimentally noticed properties and behaviors of polymers. The past few decades have seen a vast growth in the numerical simulation of structural properties which theoretically predict correlations of physical properties of the polymer systems such as the effect of the rigidity of a ring on a linear mechanically-intertwined polymer [2].

1.1 The Topics

1.1.1 Molecular Dynamics

Molecular dynamics are simulations of classical multi-particle systems [3]. The molecular dynamical systems with which we concern ourselves in this research are single polymer chains (both linear and circular) and [2]catenanes.

A single polymer chain is a system of monomers where each consecutive monomer is connected with what is modeled as a spring. Depending on the system, there may be excluded volume (EV) forces or hydrodynamics interactions (HI). EV is a repulsion force for when any two monomers in a chain overlap (or get closer than a certain distance) and HI is when particles move in response to fluid motion including motion induced by the motion of the polymer, which causes internal motion to be correlated. That being said, the normalized random forces should always be present. We study polymer chains to not only learn about the affects of Brownian dynamics, but to also rediscover scaling exponents relating the physical quantities of a chain to its molecular mass. We compare our results to those previously obtained by various researchers prior to us [4–6].

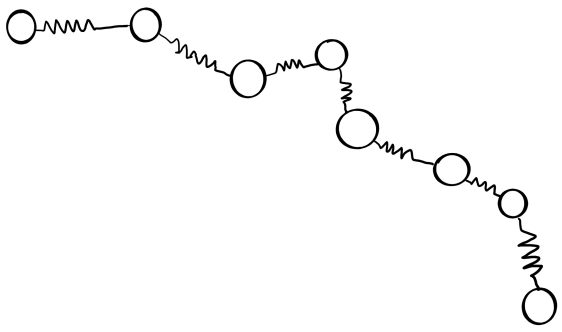
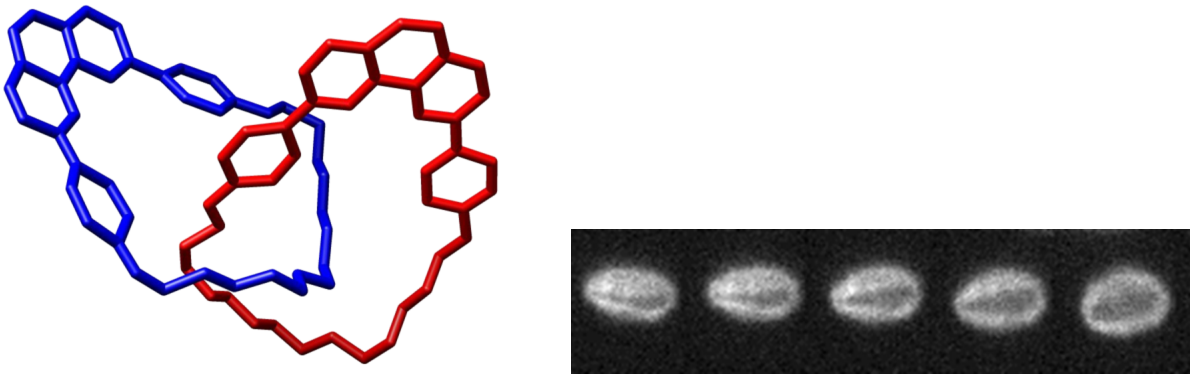


Figure 1.1: A schematic diagram of a linear polymer chain. The eight circles are beads and they are connected to their neighboring beads with springs. The springs connecting the bead are sourced by entropic elasticity [7].

Molecular topology deals with chemical compounds like polymers. If ring-shapaped polymer chains are topologically linked they form what is known as an [n]catenane.[1] The rings which form the catenane cannot be unentangled unless their covalent bonds are broken. The first catenanes were observed by American biochemist Jerome Vinograd whilst he was studying living cells [8]. Topologically linked circular molecules have since been observed in DNA particularly, at the end of replication of circular DNA.



(a) [2]catenane (b) Kinetoplast

Figure 1.2: A diagram of a [2]catenane in Figure 1.2a and one of a kinetoplast in Figure 1.2b. Images are from Wikipedia and the Klotz Lab respectively. The rings in the [2]catenane cannot be unraveled unless the covalent bonds are broken whereas the kinetoplast is a chainmail network of DNA.

A network of circular DNA within a Trypanosome parasite mitochondrion[2] which includes numerous duplicates of the complete set of mitochondrial genetic material inside a cell (i.e., a genome) is known as a kinetoplast [9]. Kinetoplasts exist in a dense, small compact particle of DNA incorporated in the large mitochondrion of Trypanosoma, a set of flagellated single-celled eukaryotes. For instance, a trypanosoma brucei parasite is a Trypanosome with a kinetoplast; the mitochondrial DNA of Trypanosome catenanes are perhaps the most prominent place in nature where catenanes are found. Trypanosomatid kinetoplast DNA (kDNA) lives vastly interlocked network of circular DNA [10].

The Klotz lab performs experiments focusing on the physics of kinetoplast DNA. To help understand the results, we are interested in simulating the equilibrium properties of linked polymers. Using techniques such as replacing solvent molecules with random forces, researchers run molecular dynamics simulations to investigate related physics such as the Rouse model. The Rouse model models the conformational dynamics of a chain with no EV but Brownian forces and a drag force is present [11]. Moreover, Rauscher, Rowan, & de Pablo (2018) simulate poly[n]catenanes by generating $n\Upsilon$ interacting beads with equal mass and diameter in n rings, each ring consisting of Υ beads along an axis "of the simulation box at distances/orientations ensuring an interlocking structure." Their consecutive beads are connected via a finitely extensible non-linear elastic (FENE) potential, and using a Verlet method their equations of

[1] n is the number of interlocked rings.
[2] A mitochondrion is a double-membrane-bound organized structure inside a living cell or cells (organelle) found in the majority of organisms where the cell's genes are chromosome-DNAs seated in different nucleus (eukaryote).

motion were integrated for certain number of time-steps. Rauscher et al. (2018) are interested in the topological effects in isolated [n]catenanes and whether HI affects catenanes or not, if the Rouse model suffices, or the Zimm model[3] is required. They find that at large length scales poly[n]catenanes do not relax much slower than isolated linear polymers and are less strongly impacted by increased segmental stiffness.

To determine the effects of the molecular mass of a polymer, an over-damped Brownian dynamics model is usually used to run and average numerous trials of simulation; Rauscher et al. (2018) concluded that inertia has no effect on a basic Brownian ideal chain. Moreover, although not taken into account by Rauscher et al. (2018), hydrodynamics are commonly of interest in molecular dynamics simulations and can be added to the Brownian dynamics-based equation of motion. One of the main focus points of the research presented in this letter is the effects of hydrodynamics on catenanes. We investigate these effects by running and analyzing simulations both with and without a hydrodynamics force.

1.1.2 Growing Self-Avoiding Walks

A self-avoiding walk (SAW) is a random walk on which no two sites share the same location and may get trapped. In polymer physics, SAWs are used to model EV between two segments of a molecule, and the data collection, analysis, and interpretation of SAWs can be used to make quantifiable predictions about the relationship between the properties of polymers. A growing self-avoiding walk (GSAW) on the other hand should explain the statistical nature of the growing process of polymerization. Close examination of the model reveals that a GSAW is best suited for addition polymerization where only one monomer unit is added to a growing polymer chain [13].

GSAWs are models for polymer growth; that is, polymers whose growth are faster than their relaxation times [14]. It is inevitable that a growing self-avoiding walk will trap itself in time; as an illustration, the average trapping length of a self-trapping walk on a square lattice is about 71 [13, 15]. We study GSAWs off any lattice to apply it to and learn about the discrete worm-like chain (DWLC) model—which is used to describe the behavior of polymers that are stiff with consecutive monomers pointing in more or less the same direction. The persistence length, l_{p}, expresses the resistance of a member of a system (such as a particle in a chain) to bending deformation, also known as bending stiffness. For a self-avoiding walk, l_{p} symbolizes the "memory" that the SAW has for the direction of its initial step [13].

1.2 Brownian Dynamics

Langevin dynamics is a mathematical model of the dynamics of molecular ensembles such as polymers. A practical molecular system is likely to be present in a solvent such as water. Realistically, because of the very large number discrepancy between the former and the latter, it would be extremely inefficient to simulate non-water molecules

[3] An extension of the Rouse model but *with* HI.

in water molecules themselves, as demonstrated in figures 2 & 3 of Chen (2021). Be that as it may, the solvent will cause friction, and the junctures of high velocity collisions will disturb the system. The main effects of the solvent are the viscous drag force and Brownian diffusion; Langevin dynamics does those implicitly so water does not have to be simulated. Langevin dynamics lets temperature be controlled or fixed, allowing to approximate properties of the canonical ensemble. Tiny round particles—beads—in a solvent will demonstrate random motion caused by thermodynamics in a way that the mean velocity of the system as a function of time will be zero as it is random [17]. What is used to describe this motion of beads is dubbed as *Brownian dynamics*; it is a simplified version with no mean acceleration of Langevin dynamics. A salient part of polymer physics is Brownian dynamics which is essentially Langevin dynamics with no inertia. The random motion of particles in media such as water is known as *Brownian motion*.

As previously mentioned, we are interested in the topics of polymer chains and growing self-avoiding walks. Random motion is an essential part of all of it especially molecular dynamics. We use Brownian dynamics to model the motion of monomers and polymers in our simulations.

1.3 Monte Carlo Simulations

A Monte Carlo simulation is routinely used to evaluate the feasible consequences of an undetermined event. Monte Carlo methods forecast a set of outcomes reached from an approximated range of values against a set of fixed inputs. That is to say, a Monte Carlo method constructs a model of viable results by leveraging a probability distribution for any variable that is uncertain by nature. The technique is then redone numerous times, each time using distinct set of random numbers between the minimum and maximum. Once a Monte Carlo simulation is finished, it produces a span of possible results with the probability of each outcome happening.

Monte Carlo techniques are used in various fields of science, mathematics, statistics, etc. They are often used for polymer physics and here particularly, one is used for the simulation of our discrete worm-like chain model.

2 Molecular Dynamics: Single Polymer Chains

2.1 Theory

We start with Newton's second law of motion being that we wish to simulate polymer chains in water:

$$m\ddot{\vec{r}} = \sum_{\iota} \vec{F}_{\iota} \tag{2.1}$$

where of course ι runs through all the forces. We expand the RHS, as the system is submerged in a fluid subjected to an external potential and collisions with the water molecules:

$$m\ddot{\vec{r}} = \underbrace{-b_{\mathrm{d}}\dot{\vec{r}}}_{\vec{F}_{\mathrm{d}}} + \sum_{\iota\neq \mathrm{d}} \vec{F}_{\iota}(t) \tag{2.2}$$

where $\sum_{\iota\neq \mathrm{d}} \vec{F}_{\iota}(t)$ is a noise term symbolizing the collision effects with the fluid molecules, $\vec{F}_{\mathrm{d}}$ is Stoke's viscous drag force with viscous drag constant $b_{\mathrm{d}} = 6\pi\curlyvee\eta$, $\curlyvee$ being the macromolecule radius, and η is the viscosity [18]. Equation (2.2) is the Langevin equation, a stochastic differential equation modeling the motion of particles in fluid [19].

If the mass of the system is negligible compared to the drag force, we ignore the effects of inertia and ergo $m \to 0$. Letting $\sum \vec{F} \equiv \sum_{\iota\neq \mathrm{d}} \vec{F}_{\iota}$, we rewrite Equation (2.2);

$$\boxed{\dot{\vec{r}} = \frac{\sum \vec{F}}{b_{\mathrm{d}}}} \tag{2.3}$$

this formula is known as the *overdamped Langevin equation*. The sum of the forces consists of:

$$\sum \vec{F} = \vec{F}_{\mathrm{rand}} + \vec{F}_{\mathrm{ev}} + \vec{F}_{\mathrm{FENE}} \tag{2.4}$$

where $\vec{F}_{\mathrm{rand}}$ is the normalized random force (Brownian force) centered at 0 with a standard deviation of 100, $\vec{F}_{\mathrm{ev}}$ is the excluded volume force which is a Hookean-like repulsive force, and $\vec{F}_{\mathrm{FENE}}$ is the FENE spring force connecting the neighboring beads;

$$\vec{F}_{\mathrm{FENE}}(\vec{r}) = \frac{3k_{\mathrm{B}}T}{l_{\mathrm{k}}}\left[\frac{\vec{r}/L_{\mathrm{s}}}{1-(\vec{r}/L_{\mathrm{s}})^2}\right] \tag{2.5}$$

where l_{k} is the Kuhn length and L_{s} is the spring length when fully stretched [20, 21]. The Kuhn length is a theoretical concept where a real polymer chain is considered as an assembly of 'Kuhn segments;' all segments are modeled to be freely-jointed with each other, the length of a fully stretched chain is then the 'number of Kuhn segments' $\times\, l_{\mathrm{k}}$. As Michael Cross remarks in his lecture notes: "The Kuhn length gives the effective link length of the polymer needed to regain the ideal polymer results."

2.2 Methods & Simulations[4]

Using PYTHON 3, we orient any number of 2D beads either horizontally or circularly with each bead having radius $\curlyvee = 0.04$. In *all* of our simulations we set: $b_{\mathrm{d}} = 1$, $L_{\mathrm{s}} = 1$, $l_{\mathrm{k}} = 0.1$, and $k_{\mathrm{B}}T = 1$. After we tell the program what

[4]The entire code and accompanying scripts for this section can be found at: `https://github.com/spyderkam/Brownian-Motion`.

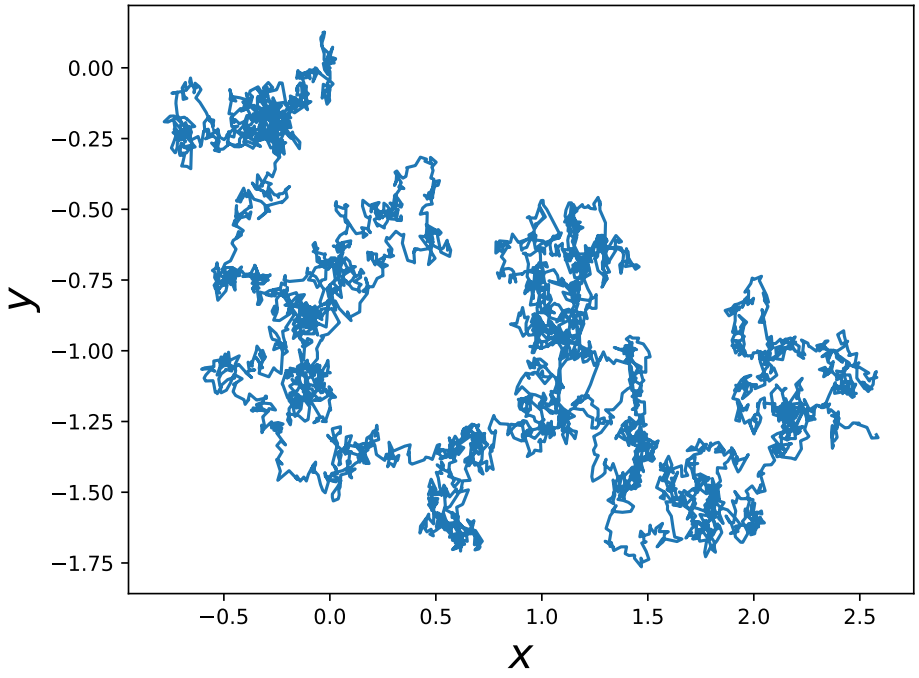


Figure 2.1: A simulated path of a two-dimensional Brownian bead. The starting position is the origin; i.e., $(x,y) = (0,0)$. The trajectory is a *random walk*. To be even more specific, the path is described by Equation (2.3); here, the drag coefficient is $b_{\rm d} = 1$ and since there are no neighboring beads, $\vec{F}_{\rm ev} = \vec{F}_{\rm FENE} = 0$ in Equation (2.4). There are $N = 5{,}001$ iterations and each time-step length is $\Delta t = 0.02$. For a large enough amount of time the path is circular.

number of beads per chain to generate, it advances the chain N time-steps each of length Δt using Equation (2.3); i.e., $\vec{r}_{\iota+1} = \vec{r}_{\iota} + \dot{\vec{r}}_{\iota}\Delta t$, where ι runs from 0 to N here. (See Figure 2.1 for an illustration of Brownian motion.) Furthermore, the program also calculates the radius of gyration after each iteration. Generally speaking, the radius of gyration, $R_{\rm g}$, is: "The distance from the center of mass of a system at which the entire mass can be concentrated without changing its moment of rotational inertia about an axis through the center of mass" [23]. We are interested in this physical quantity because for a polymer solvent it will depend on the molecular mass of the polymer chain; the expression for it at time-point ι reads:[5]

$$\left(R_{\rm g}^2\right)_{\iota} = \frac{\sum_{\jmath} m_{\jmath}\left[\vec{r}_{\iota\jmath} - \left(\vec{r}_{\rm CoM}\right)_{\iota}\right]^2}{\sum_{\jmath} m_{\jmath}} \tag{2.6}$$

where $\jmath$ runs through all the beads in the chain; we let $m_{\jmath} = 1$ so $\vec{r}_{\rm CoM} = \langle \vec{r} \rangle$ where $\vec{r}_{\rm CoM}$ is the center of mass of the chain. Over and above that, we do these Brownian dynamics based simulations to reproduce known physics before moving on to the simulations of [2]catenanes.

[5]The explicit expression we use for it in our program is:

$$R_{\rm g} = \sqrt{\sum_{\mu} \sigma_{\mu}^2}$$

where σ_{μ} is the variance of the $\mu^{\rm th}$ coordinate's data and here $\mu = x,y$ as the program is two-dimensional.

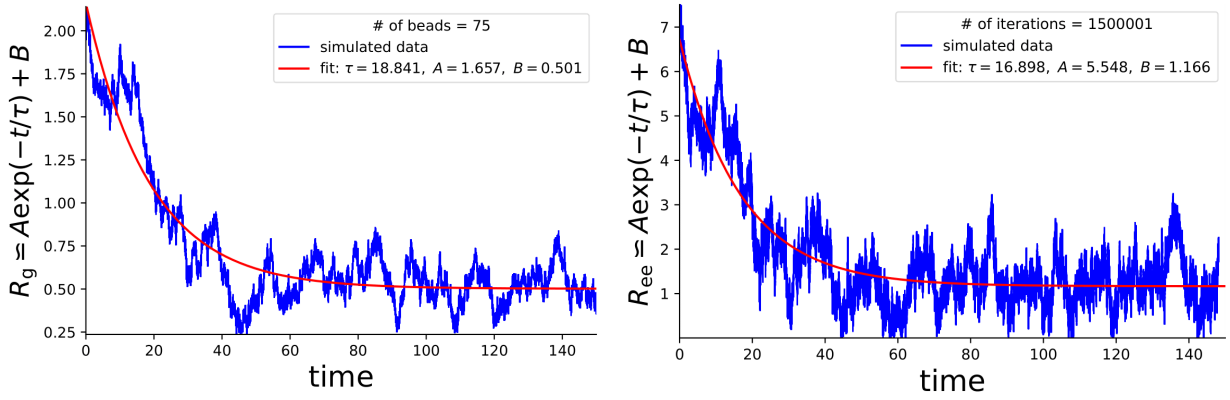
2.2.1 Linear Ideal Chain

An *ideal* (or *freely-jointed*) chain is the most basic polymer chain with no interactions between its non-adjacent assembling beads. There is no EV.

For the linear chains, starting at the origin in a Cartesian coordinate system, we place the center of each consecutive bead a distance $\Delta x = 0.09$ to the right of its previous bead. Not applying any EV, we run our simulations of a freely-jointed chain with $\Delta t = 0.0001$ and $N = 1{,}500{,}001$ for various different number of beads per chain. The overdamped Langevin equation is then iterated with the Euler method. Moreover, in addition to having the radius of gyration

Figure 2.2: The initial configuration of a polymer chain with 75 beads before applying the overdamped Langevin equation from line (2.3). Please see Figure 2.5 for this same chain stretched out both with and without EV.

automatically calculated, we calculate only for our ideal chain the *end-to-end radius* (R_{ee}) of our simulated chain at each point in time and plot both R_g and R_{ee} against time (see Figure 2.3 for reference).



(a) R_g vs t

(b) R_{ee} vs t

Figure 2.3: Sample plots of both radius of gyration and end-to-end radius vs time for a basic ideal chain (both plots are from the same simulation). In order to extract the relaxation times for R_g and R_{ee}, τ_g and τ_{ee} respectively, we fit the curve given from Equation (2.9) to our simulated data. Note that after 1 relaxation time the radius of gyration drops to $R_g(t = \tau) \approx A_g/e$. The exact expression is actually $A_g/e + B_g$ but since the equilibrium radius of gyration, B_g, seems to always be small it can approximately be disregarded in this context. The same idea also applies to the end-to-end radius, $R_{ee}(\tau) \approx A_{ee}/e$.

2.2.2 Linear Real Chain

A *real* (or *swollen*) chain is a chain where, unlike the ideal chain, inter-particle interactions exist. And so we do take into consideration F_{ev} from line (2.4).

Let the distance between bead α and bead β in the chain be $\Delta r_{\alpha\beta}$; the excluded volume force is:

$$\left|\vec{F}_{\text{ev}}\right| = \begin{cases} k_{\text{ev}}\left(\dfrac{L_{\text{s}}}{\chi} - \Delta r_{\alpha\beta}\right) & \text{if } \Delta r_{\alpha\beta} < \dfrac{L_{\text{s}}}{\chi} \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

where k_{ev} is the excluded volume constant (repulsion constant) and $\chi \equiv 3/2^{7/6}$ [24]. For *all* of our simulations of *swollen chains*, we set $|k_{\text{ev}}| = 200$ while the rest of the parameters are the same as before. Since we are using a strong constant, we expect EV to overcome the effects of our spring force as $\vec{F}_{\text{EV}} \gg \vec{F}_{\text{FENE}}$. It is interesting to

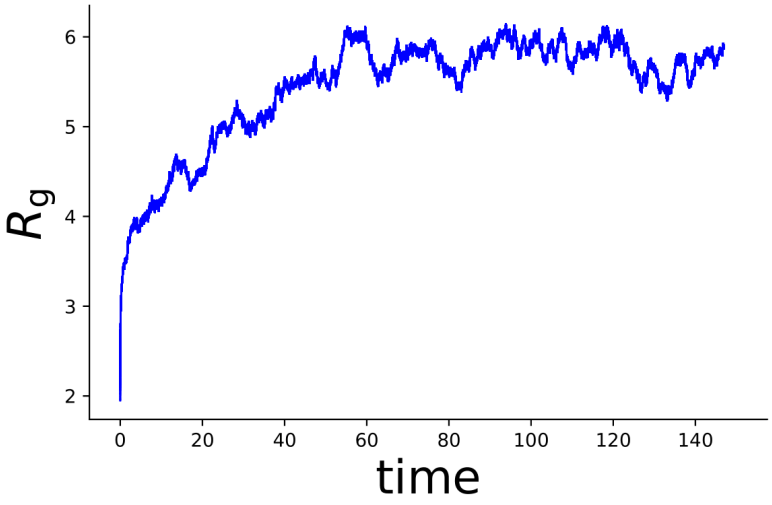


Figure 2.4: Sample plot of radius of gyration vs time for a swollen chain with no hydrodynamics. Both this plot and Figure 2.3a simulate a chain length of 75 beads and $N = 1{,}500{,}001$. Unlike before with no EV, this time R_{g} increases once relaxed; this is because we initialize the beads closer than their equilibrium length apart.[6] Moreover, for accuracy, we calculate τ_{g} here using the autocorrelation coefficient as apposed to fitting f_1 from line (2.9) to the data.

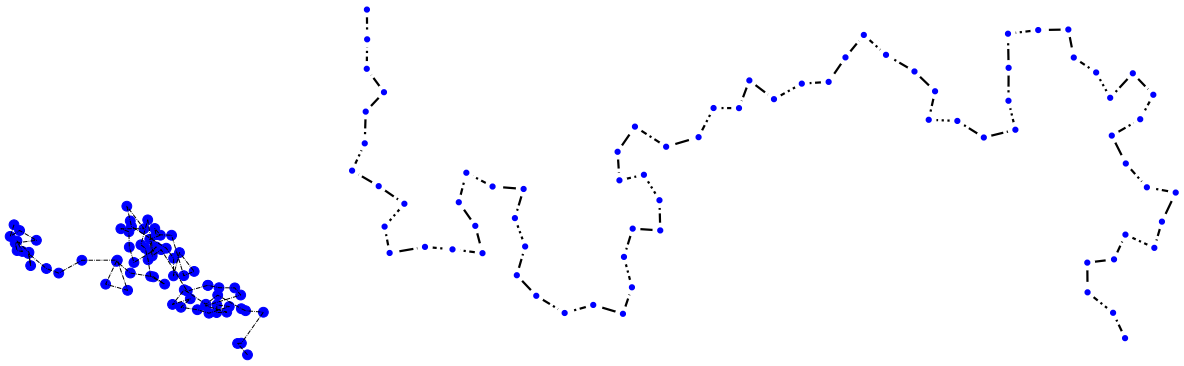
see the difference of the so-called *final* configuration of chains with and without EV, which is why we have made a side-by-side comparison in Figure 2.5.

2.2.3 Polymer Ring

After analyzing the linear chain both with and without EV (no HI in either cases), we shift our focus to a ring chain with EV and no HI. A *polymer ring*[7] is a chain whose first and last bead are connected. Without a strong excluded volume constant, in two dimensions the scaling exponents for a polymer ring should not be remarkably different than that of the linear chain [5].

We focus on a ring whose configuration at $t = 0$ is circular. To initiate the circular chain, we start by finding the

[6] There is no law that says the simulation has to start overstretched.
[7] As an illustration, bacterial DNA is circular.



(a) Ideal Chain (b) Real Chain

Figure 2.5: Freely-jointed 75-bead chain in Figure 2.5a and swollen chain of same length in Figure 2.5b. Both chains start out with the exact same initial configuration and the only difference is that the stretched real chain has $|k_{ev}| = 200$ as apposed to the ideal chain with no EV. Please note that relative to each other, the two chains are *almost* drawn to scale, the bead radii should be graphed identical for the chains to scale precisely relative to themselves.

radius of the initial configuration of the circular chain which we denote by ρ. We use δ_j as the linear distance between the center of any bead and its next consecutive neighboring bead[8] and ψ_j as the angle of each bead measured from the positive x-axis (please note that j runs from 0 to `nbeads` -1 where of course `nbeads` is the number of beads in simulation). We define $\psi \equiv \frac{2\pi}{\texttt{nbeads}}$; to initialize, we set all $\delta_j = \delta \equiv 0.09$ at $t = 0$. Thus, the *original* angle of each bead is then $\psi_j = j\psi = \frac{2\pi j}{\texttt{nbeads}}$. From basic trigonometry and the small angle approximation, $\delta_0 \simeq \rho\,\psi_1$; see Figure

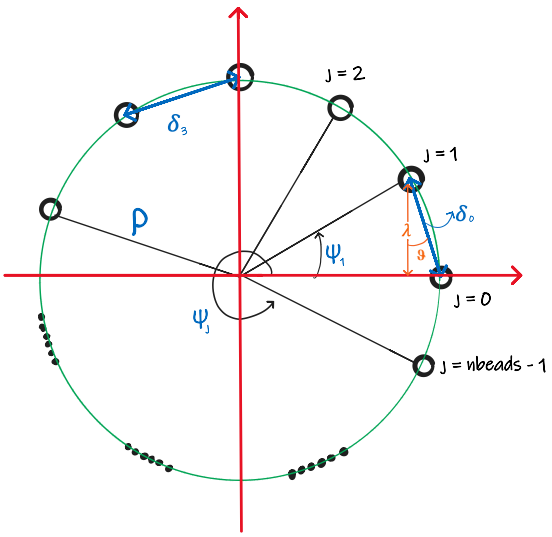


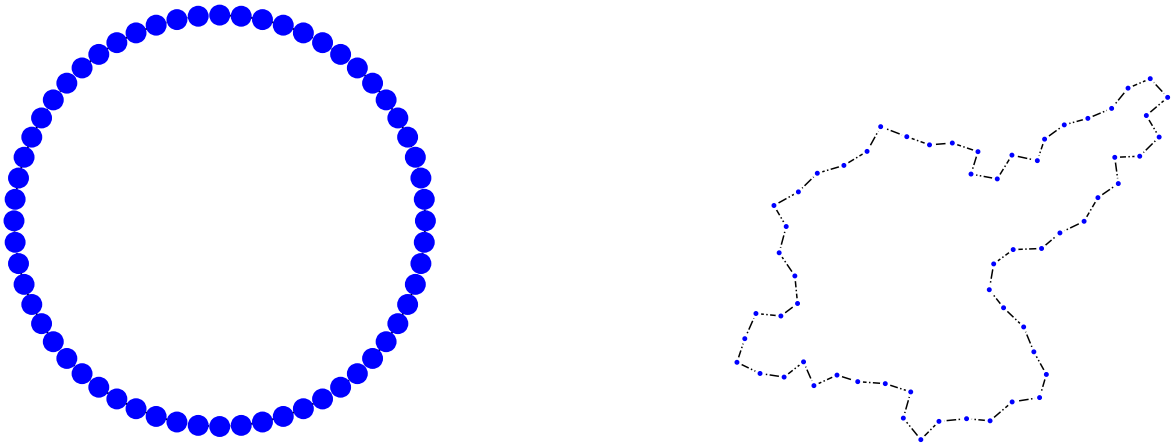
Figure 2.6: Geometrical diagram of a two-dimensional polymer ring centered at the origin. The position of each bead is $(x_j, y_j) = (\rho\cos\psi_j, \rho\sin\psi_j)$. Here in this figure, angle ϑ is only defined for the initial configuration of the polymer ring and is assumed to be small. So then originally, at $t = 0$: $\rho\sin\psi_1 = \rho\sin\psi \simeq \rho\,\psi \Rightarrow \rho\,\psi \simeq \lambda = \delta_0\cos\vartheta = \delta\cos\vartheta \simeq \delta$.

[8]For example, δ_j would be the distance between the j^{th} bead and bead number 0 in the chain.

2.6 for reference. Recall that at $t = 0$, $\delta_j = \delta$; solving for the initial configuration radius:

$$\rho \simeq \frac{\delta}{\psi} = \frac{\delta \times \texttt{nbeads}}{2\pi} \tag{2.8}$$

Given the number of beads, we can initialize a swollen polymer ring with radius given from Equation (2.8); we do this with all the same parameters as the previous section but $N = 2{,}000{,}001$ this time.



(a) Initial Configuration

(b) So-called Final Configuration

Figure 2.7: The initial configuration of a polymer ring with 60 beads in Figure 2.7a and the same ring stretched out as a result of the overdamped Langevin equation given from (2.3) in Figure 2.7b. The subplots are *not* drawn to scale relative to each other, or else the beads would be the same size. On that account, the beads in Figure 2.7a are much closer to each other than that of Figure 2.7b... From Equation (2.8), the radius of the circle in Figure 2.7a is $\rho = 2.7/\pi$.

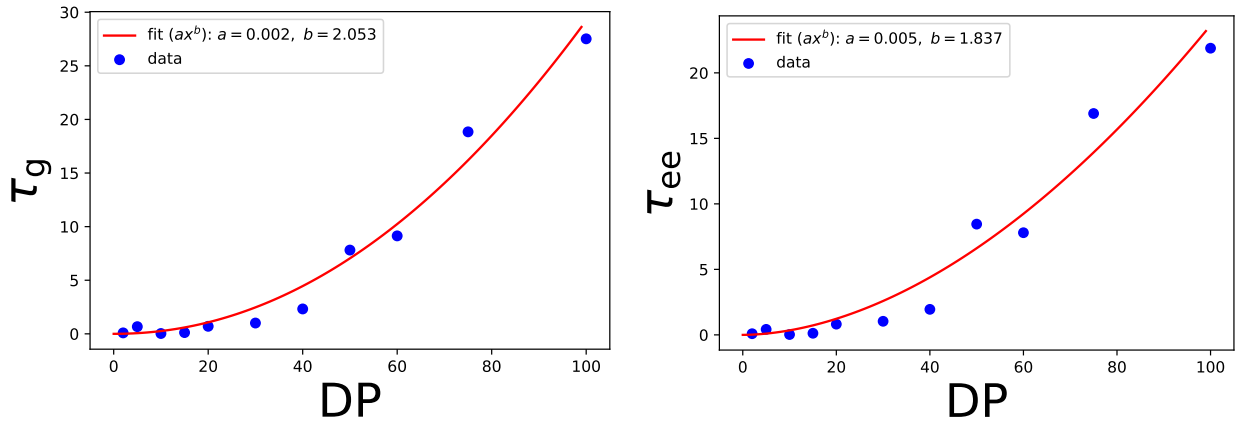
2.3 Analysis & Results

We start the analysis in the same order that we conducted our simulations, by fitting the following exponential decay function to the plots of radius of gyration as well as end-to-end radius vs. time;

$$f_1(t) = Ae^{-t/\tau} + B \tag{2.9}$$

where τ is the *relaxation time*, which along with both A and B it is determined by the fit; B is the equilibrium radius of gyration and $A + B$ is the initial radius of gyration. For the linear ideal chain, we expect the results to be quite similar, and we measured R_{ee} only for that chain. Once the relaxation time has been determined for different chains with different number of beads, we plot τ vs. the chain's molecular mass. Our springs are massless, in view of the fact that we let the mass of each bead be 1, the *degree of polymerization* (DP) of the chain would be equal to the number

of beads in the chain.

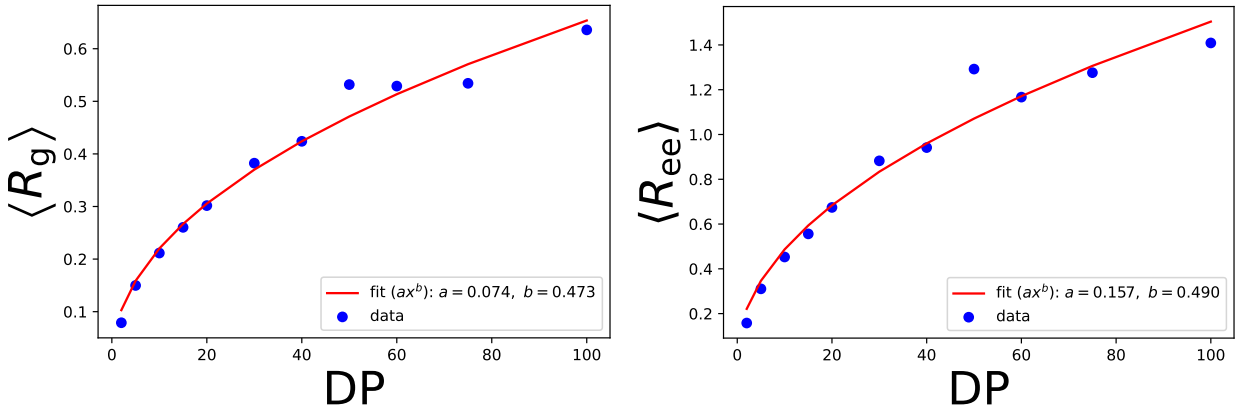


(a) τ_g vs. DP

(b) τ_{ee} vs. DP

Figure 2.8: Plots of relaxation time vs. degree of polymerization for the ideal linear chain. Using the non-linear least squares method, we fit the basic power function given from Equation (2.10) which helps us find the scaling exponents.

Furthermore, after the chain has been relaxed,[9] we focus on the R_g and R_{ee} data and plot them vs. the chain length, which is expressed using DP.



(a) R_g vs. DP

(b) R_{ee} vs. DP

Figure 2.9: Plots of the mean radius of gyration and the mean end-to-end radius against the degree of polymerization for an ideal linear chain with f_2 fitted to the simulated data.

After fitting the power function,

$$f_2(x) = ax^b \tag{2.10}$$

to the scatter plots in the subplots of Figure 2.8 and Figure 2.9, we obtain the power laws in Table 2.1. The results in

[9] For example, the horizontal part of the R_g and R_{ee} vs. time plots is when the chain is relaxed. Refer to Figure 2.3.

Table 2.1: Various Scaling Exponents for our Linear Ideal Chain Model

f_2(DP)	SIMULATION RESULTS	ROUSE THEORY
R_g	0.47 ± 0.03	0.5
τ_g	2.1 ± 0.2	2.0
R_{ee}	0.49 ± 0.04	0.5
τ_{ee}	1.8 ± 0.2	2.0

this table are consistent with the predicted values of the Rouse model [5].

As for the linear swollen chain, we use the same method to find the scaling exponent for R_g vs. DP, as shown in Figure 2.10. We then calculate τ_g by fitting Equation (2.9) to the plot of *autocorrelation coefficient*[10] vs. time. We

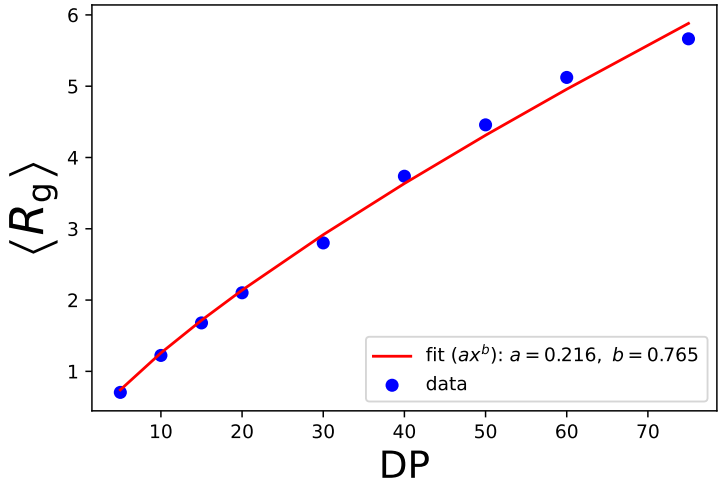


Figure 2.10: Graph and fitted curve to the averaged radius of gyration vs. molecular mass (or DP in this case) for a real linear chain.

then go on to plot τ_g against the molecular mass on a log-log scale to find the scaling exponent, which would be the slope, as shown in Figure 2.11.

We summarize our results for the swollen linear polymer chain in Table 2.2. The scaling exponent for $\langle R_g \rangle$(DP) is consistent with the Flory exponent, which is given by [6]:

$$\nu_F(\gamma) = \frac{3}{\gamma+2} \tag{2.11}$$

where γ is the number of dimensions. Our scaling is exponent for τ_g(DP) is also consistent with the expected results, which is $1+2\nu_F$ [25].[11]

[10] See Appendix B for the definition of autocorrelation.
[11] If HI was included, which it is not, this would be $1+3\nu_F$ instead [6].

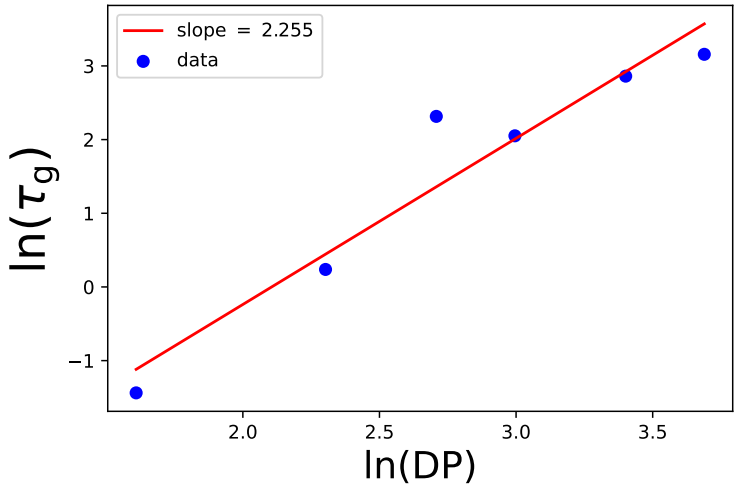


Figure 2.11: $\ln(\tau_g)$ vs. $\ln(\mathrm{DP})$; A linear fit has been included to extract the scaling exponent for $\tau_g(\mathrm{DP})$.

Table 2.2: Scaling Exponents for our Linear Real Chain Model

$f_2(\mathrm{DP})$	SIMULATION RESULTS	FLORY THEORY
R_g	0.77 ± 0.03	0.75
τ_g	2.2 ± 0.3	2.5

Last but not least in this section, we shift our analysis to the swollen circular chain by once again finding the power law for $\langle R_g \rangle$ as a function of chain length as in Figure 2.12. Our result is 0.776 ± 0.007 which indicates that our

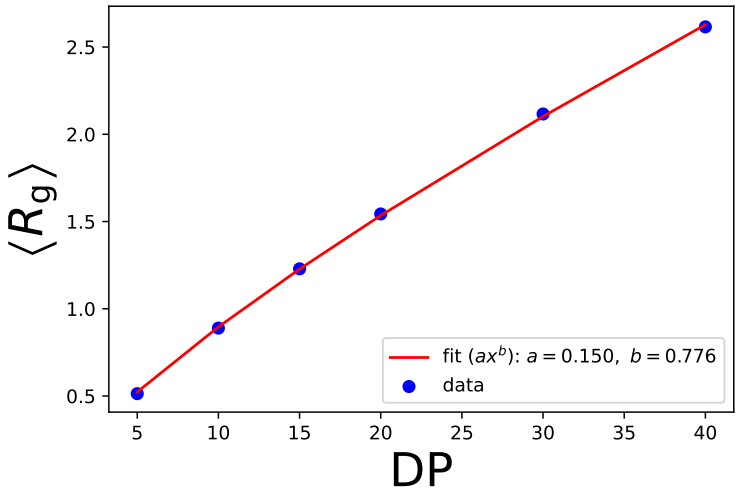


Figure 2.12: A scatter plot and fitted curve to $\langle R_g \rangle$ vs. molecular mass for a circular chain with EV.

scaling exponent for our polymer ring is consistent with the 0.77 value reported by Bishop & Michels (1986).

2.4 Summary

For single polymer chains, ideal linear, real linear, and real circular chains, we have managed to extract scaling exponents for R_g and τ_g and, in the case of the freely-jointed chain, R_{ee} and τ_{ee}. We have managed to recover the power laws relating these properties to the degree of polymerization of the chains, which is what we set out to do.

3 Molecular Dynamics: [2]Catenane

Now that we have verified known physics for single polymer chains, we set out to learn how equilibrium [n]catenanes behave. Our center of attention is linked polymers owing to the fact that we wish to better comprehend physics of kinetoplast DNA.

3.1 Model

The PYTHON 3 code written by Phil Rauscher uses Brownian dynamics to simulate ring polymer systems; specifically, poly[2]catenanes. The code uses Ermak–McCammon methods and instantaneous-averaged hydrodynamic interactions which take advantage of ring symmetries. According to the documentation of the code itself:

> The polymer model is as simple as possible: A closed loop of beads connected by Hookean springs. To model topologically linked ring polymers, the double-Rouse model constructed for poly[n]catenanes is used.

Just like the Single Polymer Chain section, we are interested in finding the scaling exponents relating the physical properties of the system to its DP. We want to know how hydrodynamic interactions affect these relationships.

3.2 Methods & Simulations

The code which simulates a [2]catenane generates a chain such that it initializes any number of even beads per ring in a three-dimensional space. The monomers are identical so we just assume the mass of each bead to be one.[12] The effective radius of each ring is given by a modified version of Equation (2.8);

$$\tilde{\rho} \simeq \frac{4 \times \texttt{nbeads}}{2\pi n} = \frac{2\Upsilon}{\pi} \tag{3.1}$$

where again `nbeads` is the *total* number of beads in the physical system and $\Upsilon = \frac{\texttt{nbeads}}{n}$ is the number of beads per ring, needless to say that $n = 2$ here. The initial bead positions (x, y, z) are then placed using Algorithm 1. Note that Algorithm 1 generates *rectangular* rings… to generate circular rings, replace $2\pi j/4$ with $2\pi j/\Upsilon$; the simulation was

[12] Which allows us to easily represent the chain length using DP.

indeed ran with rectangular rings (as originally intended by the author of the code). Anyway, once the beads are initially placed, the physics algorithm considers an HI matrix, $\underline{H}$, whose elements are tensors that set out "direction-dependent" hydrodynamics coupling between beads α and β. Then considering the center of mass of the rings, the position of the chain is updated by splitting it into time complexity $\mathcal{O}(n\Upsilon)$ for Fourier modes $\omega > 0$ and $\mathcal{O}(n^2)$ for $\omega = 0$. Please see Rauscher, Rowan, & de Pablo (2020) for the precise and in depth details of their algorithms.

Algorithm 1 Placing beads around n rings.[13]

```
for ι in range n do
    for ȷ in range ϒ do
        append 2ι to x-positions
        append ρ̃ cos(2πȷ/4) to y-positions
        append ρ̃ sin(2πȷ/4) to z-positions
    end for
end for
```

Once the beads have been placed and the [2]catenane has been constructed, the simulation proceeds using an overdamped Langevin equation; there is no option for an excluded volume force but an optional hydrodynamics force is included, as previously mentioned. Since we are interested in the effects of HI on the chain, we run the simulation twice, once without HI and once with it. Both set of simulations are run with $N = 1{,}000{,}000$ iterations and afterwards we set $\Delta t = 0.01$ as the simulation itself does not consider time-step 'lengths' for iterations. Due to the lack of computational power at our disposal, we only run simulations up to $\Upsilon = 30$. The parameters of interest are: the radius of gyration (R_g), the relaxation time (τ_g), and the diffusion coefficient ($\mathcal{D}$). Our mission is to see how they depend on hydrodynamics.

3.3 Analysis & Results

3.3.1 The Average Radius of Gyration

We start with the radius of gyration, or to be specific, the mean of it, $\langle R_g \rangle$. Plotting it against the chain length for both no-HI and full-HI gives virtually the same outcome, as seen in Figure 3.1.

3.3.2 The Relaxation Time

To deduce the relaxation times, we use the same exact method as in Section 2.3; i.e., by utilizing our autocorrelation function, `AC`, from Appendix B.2. We then analyze this data with Figure 3.2.

[13] The indexing in this algorithm starts from 0. (As well as Algorithm 2 for that matter.)

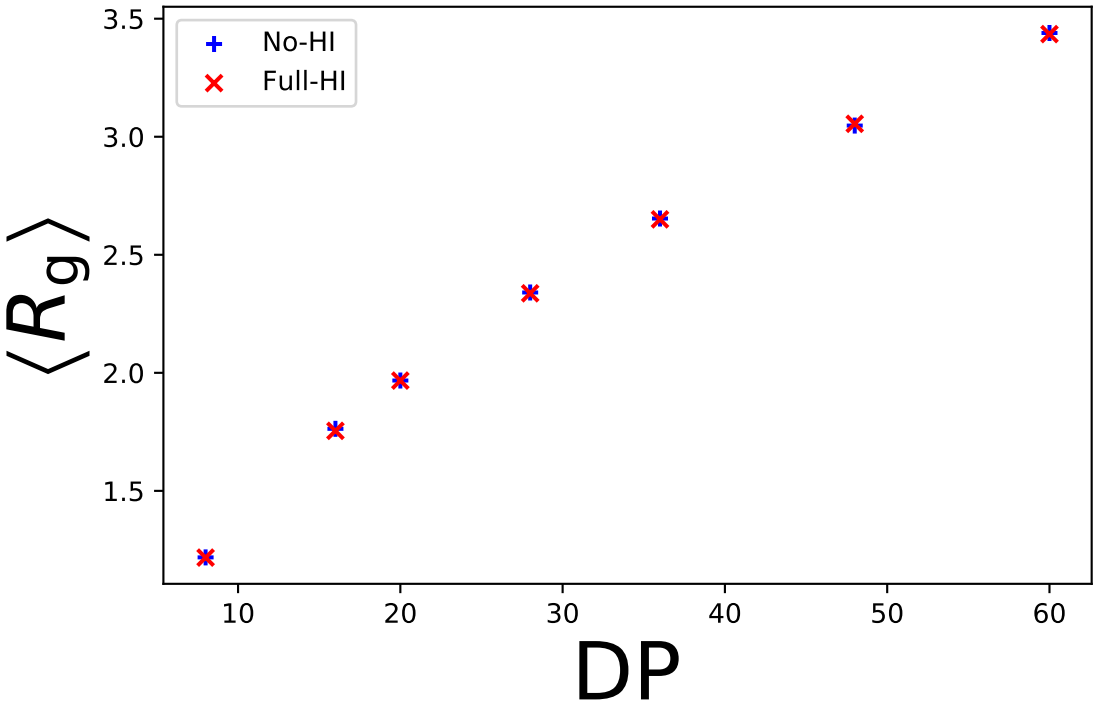
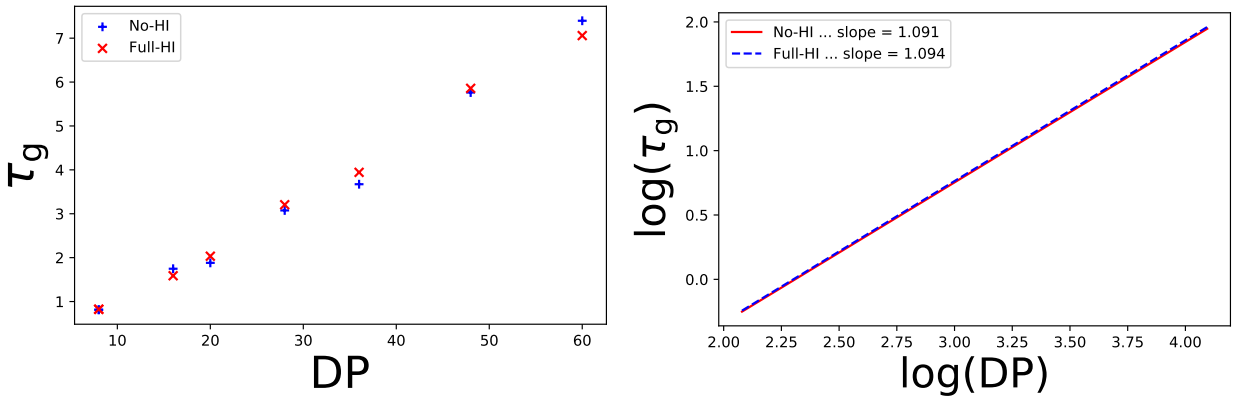


Figure 3.1: $\langle R_g \rangle$ vs DP; it is quite clear that HI has no effect whatsoever on the radius of gyration here.



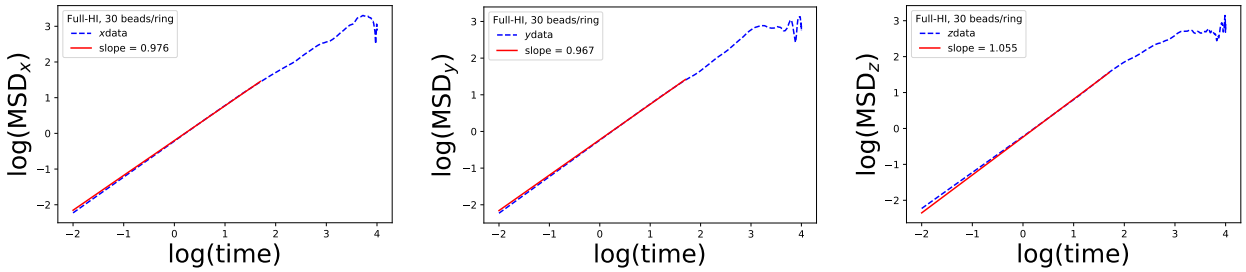
(a) τ_g vs DP

(b) $\log(\tau_g)$ vs log(DP) linear fit

Figure 3.2: The relaxation time plotted against the degree of polymerization in Figure 3.2a and a linear fit to the same simulated data on a (natural) logarithmic scale in Figure 3.2b. From these plots, it appears that the hydrodynamics forces have very little, if any, effect on the relaxation times of three-dimensional [2]catenanes with chain lengths up to 60.

3.3.3 The Diffusion Coefficient[14]

To find the diffusion constant, we start by finding the *mean square displacement* (MSD). The dimensional components of the MSD should scale linearly with time (as in Figure 3.3).



(a) $\log(\text{MSD}_x)$ vs t (b) $\log(\text{MSD}_y)$ vs t (c) $\log(\text{MSD}_z)$ vs t

Figure 3.3: The dimensional components of MSD plotted against time on a log-log scale for $\Upsilon = 30$. Although the portion of data that we fit a line to does not *exactly* have a slope of unity in this particular example, it is close enough as the slopes ≈ 1. We then use this same exact portion of data to find the total MSD. (Please note that the linear curve has been fitted to the same portion of data for all components.)

Moreover, we note that the total MSD is equal to the sum of its components as well as its relationship with $\mathcal{D}$:

$$\text{MSD} = \sum_{\iota} (\text{MSD})_{\iota} \tag{3.2a}$$

$$\text{MSD} = 2\gamma\mathcal{D}t \tag{3.2b}$$

whereas before, γ is the number of dimensions (in this case $\gamma = 3$) and ι is the dimension itself. Putting them together:

$$\sum_{\iota} (\text{MSD})_{\iota} = 2\gamma\mathcal{D}t \tag{3.3}$$

After adding the relevant parts of MSD_{ι} together, we plot it against time; expecting it to be linear, we then divide the slope by 6 to find the diffusion coefficient. We do this process for various different number of beads per ring (up to $\Upsilon = 30$), for both no-HI and full-HI, and plot the results against the degree of polymerization in Figure 3.5.

3.4 Epilogue

Although we were not expecting hydrodynamics to affect the radius of gyration as it is a static measurement, we were expecting HI to have at least a noticeable effect on the relaxation time as well as the diffusion constant; we expected τ_g to decrease with time when HI is included but $\mathcal{D}$ to be more with HI.

Both with and without hydrodynamic interactions, we obtain a slope of roughly 1 for $\log(\tau_g)$ vs. $\log(\text{DP})$, but

[14] See Appendix A for the definition of diffusion.

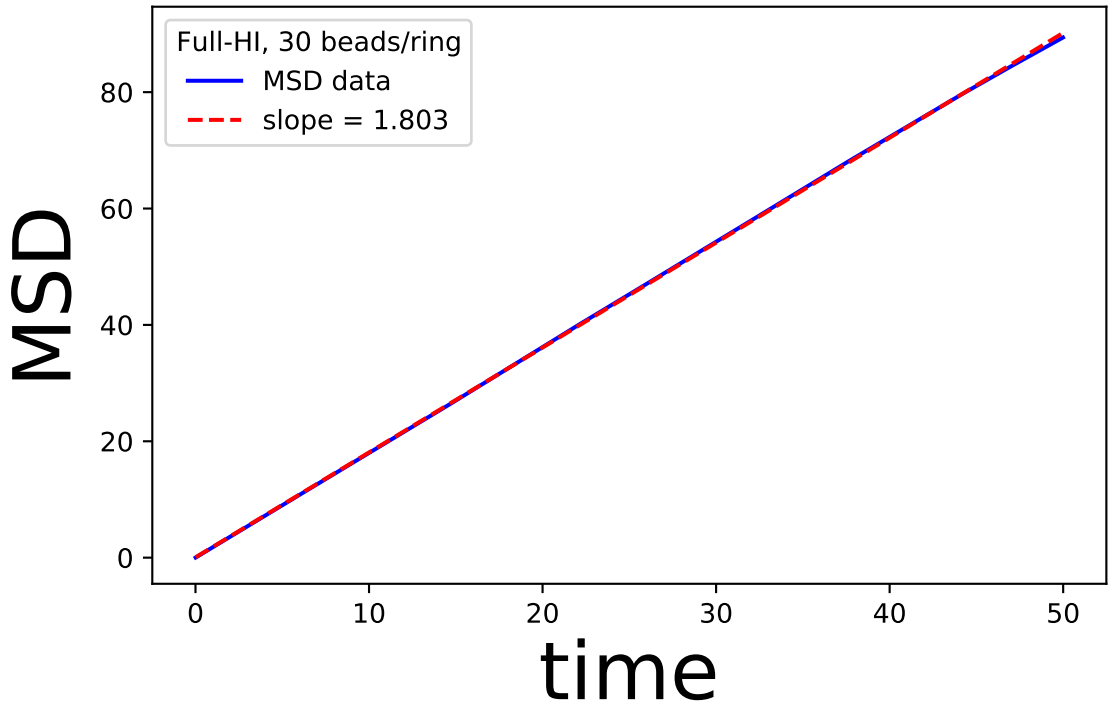
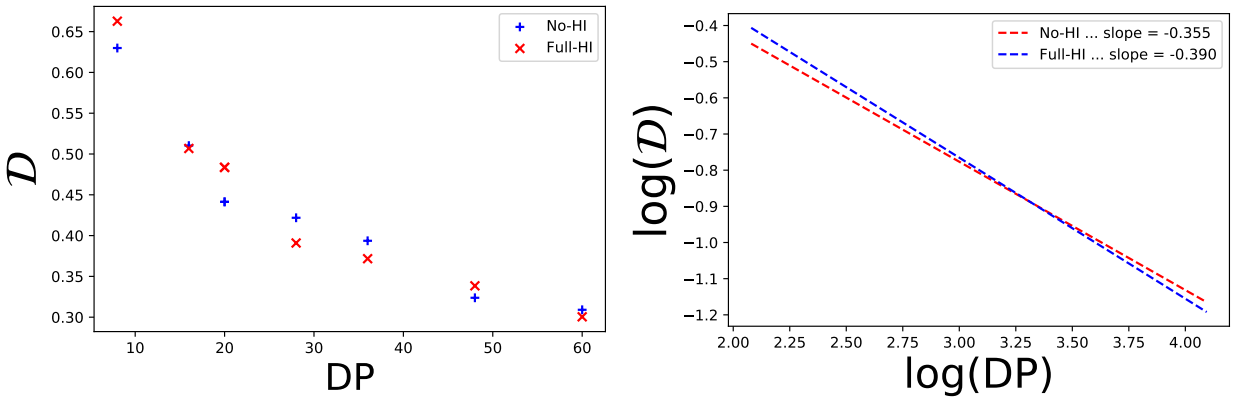


Figure 3.4: A sample plot of the relevant parts of $\sum_t \text{MSD}_t$ vs time; see Figure 3.3 for reference. The plot is linear as expected; to find $\mathcal{D}$, divide the fitted slope by 2γ.



(a) $\mathcal{D}$ vs DP

(b) log($\mathcal{D}$) vs log(DP) linear fit

Figure 3.5: The diffusion coefficient plotted against the degree of polymerization in Figure 3.5a and a linear fit to the same simulated data on a (natural) logarithmic scale in Figure 3.5b. Apart from some small fluctuations, the hydrodynamics forces do not seem to have much of an effect at these chain lengths.

using this very code, Rauscher et al. (2020) predict a slope of 1.5 for full-HI. The maximum Υ used in their research was 96 as opposed to our 30, more than 3 times the amount. We believe that the discrepancy in our results is due to our short chain length, as even $\Upsilon = 96$ is relatively short, according to Rauscher et al. (2020). By that same token, Tree, Muralidhar, Doyle, & Dorfman (2013) allude that shorter chain lengths might be not good models for polymers. We also believe that when $n = 2$, HI has a much less noticeable effect on the $[n]$catenane as opposed to larger n, and the results here for our [2]catenane reaffirm that.[15]

4 Trapping Statistics

4.1 Prologue

This section of the thesis is separate from the molecular dynamics section because it focuses on the idea of growing self-avoiding walks. It carries on with the research done by Hooper & Klotz (2020). The GSAW model has not been evaluated off a lattice. The DWLC model is the most general polymer model that is not on a lattice; we simulate a growing self-avoiding walk.

4.2 Theory

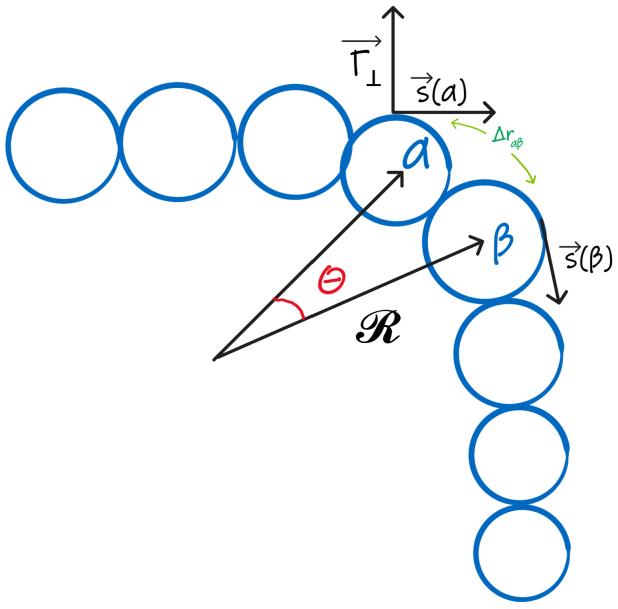


Figure 4.1: A schematic, geometrical diagram of a worm-like chain model. α and β represent beads, $\Delta r_{\alpha\beta}$ is the contour distance between their centers, $\mathcal{R}$ is the local radius of curvature, the $\vec{s} = \partial\vec{\Gamma}_\perp/\partial\vec{r}$ vectors are the tangent vectors, $\vec{\Gamma}_\perp$ is the normal unit vector, θ is the *random* bending angle which for the DWLC should be small.

[15] The Kotz lab works with poly$[n]$catenanes for up to $n = 5{,}000$!

A basic worm-like chain (WLC) models polymers that are semi-flexible, fairly stiff with successive segments pointing in about the same direction, and essentially with $l_p \sim$ the polymer length. For a WLC, $l_p = \frac{1}{2} l_k$. Please see Figure 4.1 for our WLC model, which we use as a prototype for our DWLC which grows by adding consecutive beads to the chain. We let l_p be defined by the tangent correlation function $g(\Delta r_{\alpha\beta}) = \langle \vec{s}(\alpha) \cdot \vec{s}(\beta) \rangle = \exp\left[-\left|\Delta r_{\alpha\beta}\right| / l_p\right]$, which can be written as $\left\langle \cos[\theta(\Delta r_{\alpha\beta})] \right\rangle$ [28],

$$\langle \cos\theta \rangle = e^{-\left|\Delta r_{\alpha\beta}\right| / l_p} \tag{4.1}$$

Assuming 3D space where the chain is bending only in a 2D plane, and bending θ is small:

$$\theta \simeq \frac{\Delta r_{\alpha\beta}}{\mathcal{R}} \tag{4.2}$$

We can then use a bending energy expression analogous to that of the elastic potential energy [28];

$$V(\theta) \simeq \frac{1}{2\Delta r_{\alpha\beta}} K\theta^2 \tag{4.3}$$

where K is the bending stiffness. Using the following partition function Z in spherical coordinates for the bending chain,

$$Z = \int_0^{2\pi} d\varphi \int_0^{\pi} d\theta \, \sin(\theta) \, e^{-V/(k_B T)} \tag{4.4}$$

we can find the variance of θ: $\sigma_\theta = \langle \theta^2 \rangle - \cancelto{0}{\langle \theta \rangle^2}$;

$$\langle \theta^2 \rangle = \frac{1}{Z} \int_0^{2\pi} d\varphi \int_0^{\pi} d\theta \, \sin(\theta) \, \theta^2 \, e^{-V/(k_B T)} \tag{4.5}$$

$$= \frac{2\Delta r_{\alpha\beta} k_B T}{K} \tag{4.6}$$

where we have used the small angle approximation. Now considering the Taylor expansion of the trigonometric form of the tangent correlation function:

$$g = \langle \cos\theta \rangle \tag{4.7}$$

$$= \left\langle 1 - \frac{\theta^2}{2} + \cdots \right\rangle \tag{4.8}$$

$$\simeq 1 - \frac{1}{2} \langle \theta^2 \rangle \tag{4.9}$$

side note

Recall (4.2) and so that,

$$\frac{\partial \vec{\Gamma}}{\partial r_{\alpha\beta}} \simeq \frac{d\theta}{dr_{\alpha\beta}} = \frac{1}{\mathcal{R}}$$

where of course $1/\mathcal{R}$ is the inverse of local radius of curvature.

With this and (4.6),

$$\langle \cos\theta \rangle \simeq 1 - \frac{\Delta r_{\alpha\beta} k_{\mathrm{B}} T}{K} \tag{4.10}$$

now considering the Taylor expansion of the exponential form of g:

$$e^{-|\Delta r_{\alpha\beta}|/l_{\mathrm{p}}} \simeq 1 - \frac{|\Delta r_{\alpha\beta}|}{l_{\mathrm{p}}} \tag{4.11}$$

equating the RHSs of (4.10) and (4.11), the following expression is obtained:

$$l_{\mathrm{p}} = \frac{K}{k_{\mathrm{B}} T} \tag{4.12}$$

Then, from (4.3), (4.6), and (4.12) we derive an equation directly relating the persistence length to the variance of bending angle;

$$\frac{2\Delta r_{\alpha\beta}}{\sigma_\theta} \simeq l_{\mathrm{p}} = \frac{K}{k_{\mathrm{B}} T}\,. \tag{4.13}$$

The expressions in (4.13) relate l_{p}, σ_θ, and K. When $l_{\mathrm{p}} = 0$ we have a freely-jointed walk, the larger this variable is the stiffer the chain is.

4.3 Methods & Simulations

We use a MATLAB code written by Alex Klotz for our simulations of growing self-avoiding walks in a two-dimensional space.[16] To horizontally initialize the chain, we place our first bead to be located at the origin and the second bead to be located at $(x,y) = (2\text{r},0)$ where we set the radius of the beads r to unity. The next bead in the chain is placed using Algorithm 2. Please note that `rand` is a random number from the uniform distribution in the range $[0,1)$ whilst `randn` is a random number from from the *standard normal distribution*, `chain` is the array which

[16] Our Theory is in 3D while this is 2D; thus, this is a simplified class.

Algorithm 2 Generating the next bead in a GSAW chain.

$\vec{v} =$ `chain[-1]` $-$ `chain[-2]`
`tries` $= 1$; `condition` $=$ **False**
while `tries` ≤ 100 **&&** `condition` $=$ **False do**
  if $l_p = 0$ **then**
    $\varphi = \frac{4\pi}{3}(\texttt{rand} - 1/2)$
  else
    $\varphi = \texttt{randn}/\left(\sqrt{l_p/2}\right)/\texttt{tries}$
  end if

  $R = [(\cos\varphi, -\sin\varphi),\ (\sin\varphi, \cos\varphi)]$
  $b_{\text{new}} = R \cdot \vec{v}^{\intercal}$
  for `bead` **in** `chain` **do**
    $\Delta x = b_{\text{new}}$`[0]` $-$ `bead[0]`
    $\Delta y = b_{\text{new}}$`[1]` $-$ `bead[1]`

    if $\sqrt{(\Delta x)^2 + (\Delta y)^2} > 2\text{ﾑ}$ **then**
      append b_{new} to `chain`
      `condition` $=$ **True**
      break
    end if
  end for

  `tries` $=$ `tries` $+ 1$
end while

contains the (x, y) coordinates for each bead in the chain. If the algorithm fails to generate a bead which does not overlap with any of the beads already in the chain in its first 100 tries, it stops. A diagram of a discretely growing worm-like chain is graphed in Figure 4.2.

4.3.1 First Simulation

Walks were simulated for various different rough values of l_p between 0 and 8, as well as 12.5, 18, and 32; we did this simulation for 100,000 walks. Once the simulation completes we average our results and look at parameters of interest, mainly the mean trapping length. We also look at the histograms of 'number of simulations' vs.'trapping lengths' for each approximate value of l_p. A sample of the raw histograms are plotted in Figure 4.3 for $l_p = 0,\ 8,$ and 32.[17]

4.3.2 Second Simulation

We run the simulation once more to extract the radius of gyration, R_g; however, due to the very large amount of time it took to complete the calculation of R_g^2 using Equation (2.6), only $N = 16{,}190$ walks (iterations) were completed (we believe that this is enough to analyze the radius of gyration). Once the simulation is complete, the exported data for R_g^2 is a $p \times q$ matrix whose *rows* are chains with different trapping lengths and the *columns* are different walks of our

[17] Including more histograms for other values of l_p would germ an overcrowded graph causing information to be lost.

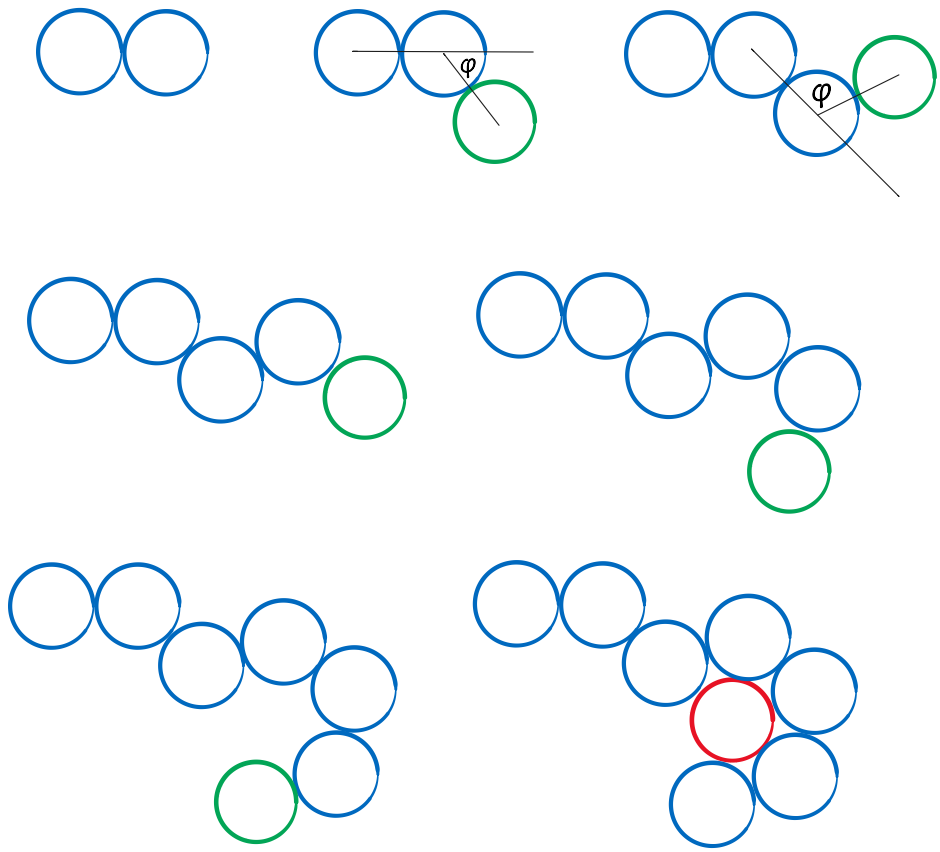


Figure 4.2: A schematic diagram of a two-dimensional DWLC. The chain start at the top left and each added bead is in green until it traps itself with the final red bead.

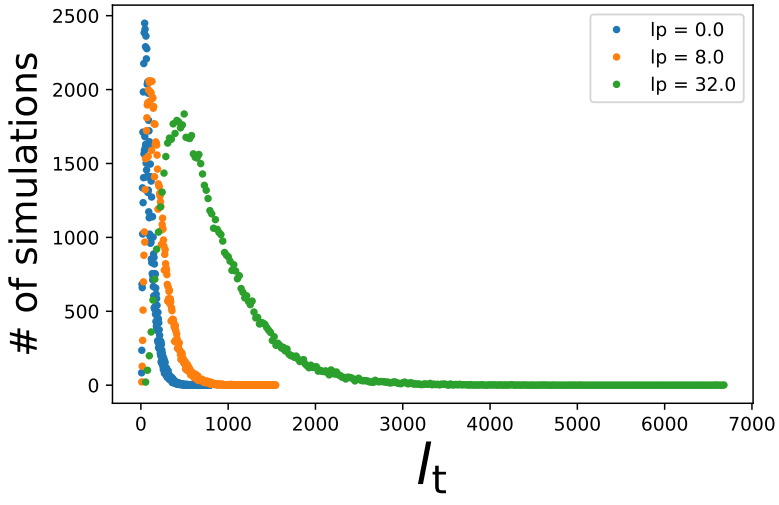


Figure 4.3: Raw histograms corresponding to 3 different values of l_p: 0, 8, and 32. The plot is difficult to analyze, thus far. Once an appropriate curve has been fitted to the data and then normalized, the histograms can be surveyed.

off-lattice GSAW. The matrix of R_{g}^2, that is R_{g}^2, where ℓ represents the length and w represents the walk is below:

$$R_{\mathrm{g}}^2 = \begin{pmatrix} w_0\ell_0 & w_1\ell_0 & w_2\ell_0 & \cdots & w_q\ell_0 \\ w_0\ell_1 & w_1\ell_1 & w_2\ell_1 & \cdots & w_q\ell_1 \\ w_0\ell_2 & w_1\ell_2 & w_2\ell_2 & \cdots & w_q\ell_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_0\ell_p & w_1\ell_p & w_2\ell_p & \cdots & w_q\ell_p \end{pmatrix} \tag{4.14}$$

Here in this simulation, $p = 669$ which is the maximum trapping length of all the simulated walks, $q = N$ which is the number of walks completed. Now that the numerical values for R_{g}^2 have been stored for each different length of each different walk, the *population average* can be found, its general formula is:

$$\text{POPULATION AVERAGE} = \frac{1}{\mathcal{N}} \sum X \tag{4.15}$$

where $\mathcal{N}$ is the number of actual items in the group[18] and X are the items themselves. In other words, we are basically purging the zero elements from R_{g}^2 before evaluating its mean.

4.4 Results

We start off by fitting a curve define by the following function to all of our histograms where the x-axis has been $\div$ the mean trapping length;

$$f_3(x) = \mathscr{A}\left(x - \frac{\epsilon}{\langle l_{\mathrm{t}} \rangle}\right)^{\mathscr{B}} \exp\left[-\mathscr{C}\left(x - \frac{\epsilon}{\langle l_{\mathrm{t}} \rangle}\right)\right] \tag{4.16}$$

where ϵ represents the minimum trapping length, we use $\epsilon = 6$; $\mathscr{A}$, $\mathscr{B}$, and $\mathscr{C}$ are parameters determined by the fit. Then we normalize our curves; we do this by taking into consideration that the total area $\propto x \cdot y$; meaning to say, to normalize the area under the curve, we divide the `ydata` by the original total area under the curve, where the total area is found by integrating.[19] We plot these normalized curves in Figure 4.4.

Furthermore, we continue by looking at how the trapping length l_{t} relates to the persistence length l_{p} by plotting their data from the first simulation against each other in Figure 4.5. We notice that the trapping length starts converging when the persistence length is at around 1 and does converge just about after $l_{\mathrm{p}} = 5$. It converges to $l_{\mathrm{t}} \approx 24.7 l_{\mathrm{p}}$. We then shift our attention to the outcome of the second simulation, which was focused on the radius of gyration. We are interested in the scaling exponent which relates the radius of gyration to the trapping length. After graphing the

[18] Caligraphy $\mathcal{N}$ not to be confused with N which is the symbol for the *number of iterations*.
[19] We used NumPy's `trapz` method, which utilizes the composite trapezoidal rule.

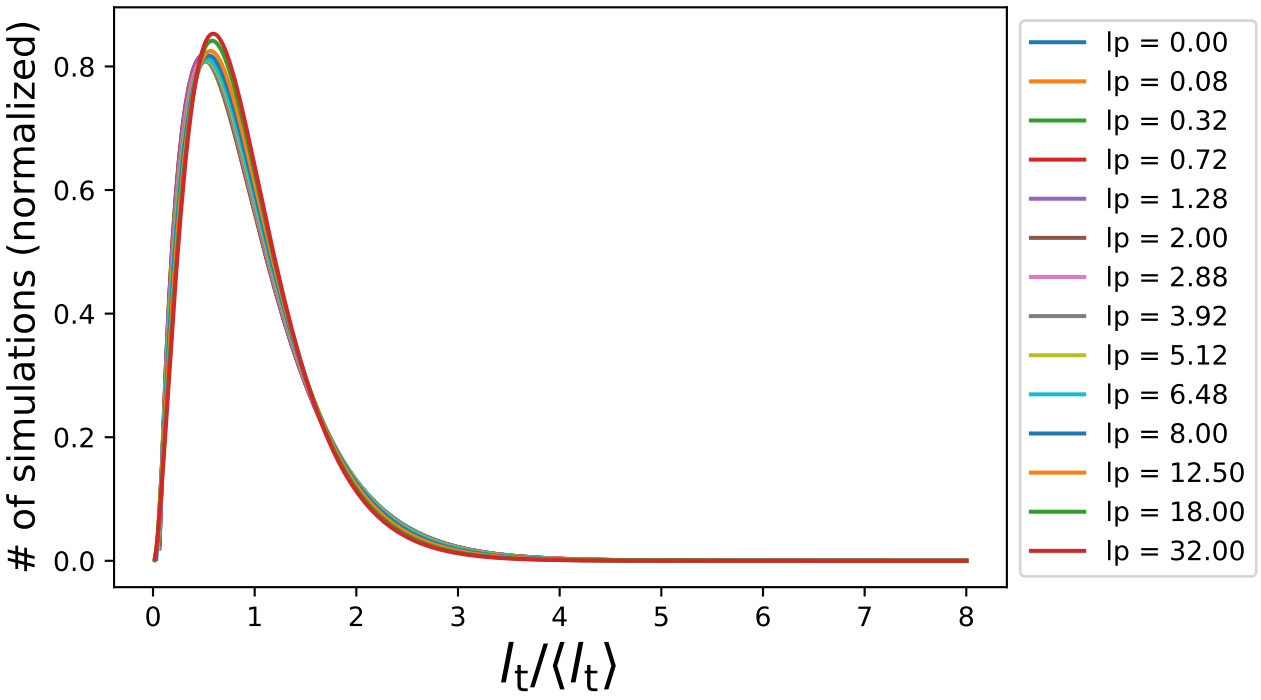


Figure 4.4: Histogram of the trapping length divided by its average for each value of l_p for the number of simulations normalized. We see that $l_t \approx \langle l_t \rangle$ for the majority of simulations, as expected (which implies that the simulations were ran successfully). It is noteworthy that for all values of l_p, the peak is at $\gtrsim 0.8$.

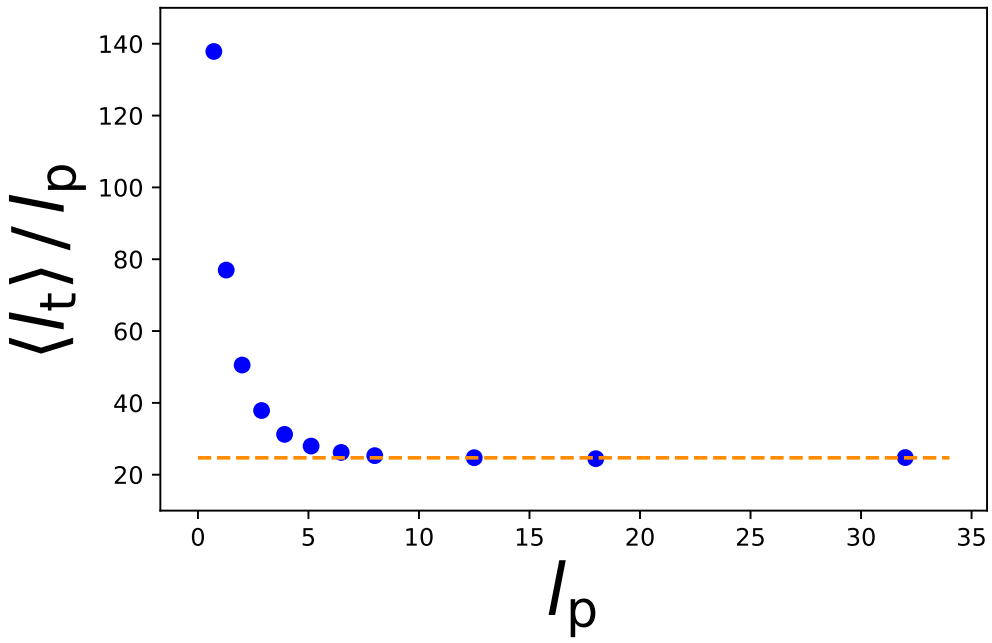


Figure 4.5: $\langle l_t \rangle / l_p$ vs. l_p for an averaged 10^5 walks. A horizontal dashed line at $\langle l_t \rangle / l_p = 24.7$ has been added to the plot to show that the trapping converges at approximately that value $\times$ the persistence length. The transition starts to happen at about $l_p = 1$ and completes just after $l_p = 5$.

$\langle R_g^2 \rangle$ related plots to the commencing curve of the graph (before the curve gets *squiggly*) we should fit the power function f_2. But the interesting quantity is $\langle R_g \rangle$ itself and not $\langle R_g^2 \rangle$. Usually, one cannot simply divide the power law by the exponent of the average radius of gyration as typically $\langle \Xi \rangle \neq \langle \Xi^\xi \rangle^{1/\xi}$. One solution would be to go back to the beginning and take the square root of all the elements of R_g^2; but, it would be more efficient to simply divide the power law found for $\langle R_g^2 \rangle (l_t)$ by 2, even though it is not the same as $\langle R_g \rangle \ldots$ This is all right because for a large amount of data they *will* be the same; the variance is $\langle R_g^2 \rangle - \langle R_g \rangle^2$ and gets smaller and smaller with more and more data.

$$\lim_{N \to \infty} \sigma^2 = 0 \tag{4.17}$$

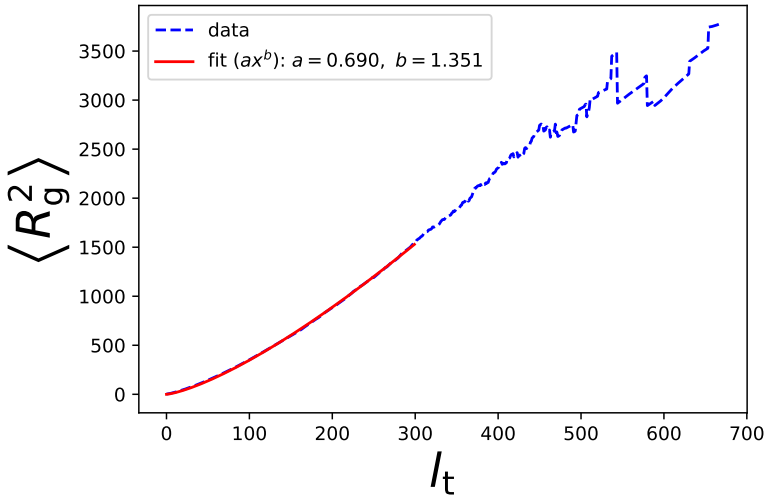


Figure 4.6: $\langle R_g^2 \rangle$ vs. l_t with the power fit. The power function (2.10) has been the fitted to the early, appropriate curve of the graph to obtain a proper fit. To obtain the scaling exponent for $\langle R_g \rangle$ vs. l_t one can simply divide the value of b here by 2, as there is a relatively large and enough amount of data.

And so the power law for $\langle R_g \rangle (l_t)$ is $\frac{b}{2} = 0.6757 \pm 0.0008$, which is consistent with the findings of Hooper & Klotz (2020) that for a GSAW, the scaling exponent R_g vs. trapping is ≈ 0.68; wherefore not a consequence of a lattice. Likely because over long distances a square lattice is not expected to make much difference compared to open space.

5 Outlook

Topics in polymer physics have computationally been researched in the fields of molecular dynamics and trapping statistics. Molecular dynamics has been split into two separate sections: Single Polymer Chains and the [2]Catenane. Single polymer chains have been simulated for both linear and circular configurations, known physics have been corroborated, and Brownian dynamics has been understood. Poly[2]catenane scaling exponent results have not been confirmed as the ultimate conclusion was that our chains were too short to properly model a polymer.

As for the Trapping Statistics section of this letter: Nearly the same scaling exponent for R_g vs. l_t has been found as was for the square lattice, verifying that the lattice has no effect on their relation. A relationship between trapping and persistence length has also been derived. All in all, using Brownian dynamics and Monte Carlo simulations, we have modeled basic polymer systems, obtained results, and drawn conclusions for the dynamics of the systems of interest.

References

[1] Britannica, *Polymer*, retrieved on 11/09/2022 from `https://www.britannica.com/science/polymer`.

[2] Pietro Chiarantoni & Cristian Micheletti, *Effect of Ring Rigidity on the Statics and Dynamics of Linear Catenanes*, American Chemical Society, (2022) **55** (11), 4523-4532, [DOI: 10.1021/acs.macromol.1c02542].

[3] K. Vollmayr-Lee, *Introduction to Molecular Dynamics Simulations*, American Journal of Physics (2020) **88** (5), 401, American Association of Physics Teachers, [DOI: 10.1119/10.0000654].

[4] Marvin Bishop & J. P. J. Michels, *Scaling in Two-Dimensional Linear and Ring Polymers*, Journal of Chemical Physics (1986) **85** (2), 1074, [DOI: 10.1063/1.451300].

[5] Chen, Zhao, Liu, Chen, Li, An, *Effects of Excluded Volume and Hydrodynamic Interaction on the Deformation, Orientation and Motion of Ring Polymers in Shear Flow*, Soft Matter Journal (2015) **11** (26), 5265-5273, [DOI: 10.1039/c5sm00837a].

[6] L. Levitov, *Polymers Notes*, Massachusetts Institute of Technology, 2003.

[7] ScienceDirect, *Entropic Elasticity*, retreived on 11/16/2022 from `https://www.sciencedirect.com/topics/engineering/entropic-elasticity`.

[8] A. Vologodskii, *Unlinking of Supercoiled DNA Catenanes by Type IIA Topoisomerases*, Biophysics Journal (2011) **101** (6), 1403-1411, [DOI: 10.1016/j.bpj.2011.08.011].

[9] Edgar J. Garcia, *Equilibrium Properties of Catenated Membranes*, master's thesis, California State University, Long Beach, October 2021.

[10] ScienceDirect, *Kinetoplast DNA*, retrieved on 15/02/2022 from `https://www.sciencedirect.com/topics/pharmacology-toxicology-and-pharmaceutical-science/kinetoplast-dna`.

[11] Greg Beaucage, *The Rouse Model*, University of Cincinnati, retrieved on 03/13/2021 from `https://www.eng.uc.edu/~beaucag/Classes/Physics/DynChapter6html/Chapter6.html`.

[12] Rauscher, Rowan, & de Pablo, *Topological Effects in Isolated Poly[n]catenanes: Molecular Dynamics Simulations and Rouse Mode Analysis*, ACS Macro Letters (2018) **7** (8), 938-943, [DOI: 10.1021/acsmacrolett.8b00393].

[13] Hooper & Klotz, *Trapping in Self–Avoiding Walks with Nearest–Neighbor Attraction*, Physical Review E (2020) **102** (3), 032132, American Physical Society, [DOI: 10.1103/PhysRevE.102.032132].

[14] J. W. Lyklema & K. Kremer, *The growing self avoiding walk*, Journal of Physics A: Mathematical and General (1984) **17** (13), L691, [DOI: 10.1088/0305-4470/17/13/003].

[15] S. Hemmer & P. C. Hemmer, *An average Self–Avoiding Random Walk on the Square Lattice Lasts 71 Steps*, Journal of Chemical Physics (1984) **81** (1), 584, [DOI: 10.1063/1.447349].

[16] Jiuke Chen, *Coarse-grained Simulations for Poly (Ethylene Oxide) Linear Chains and [2]Catenanes in Water*, master's thesis, The University of Akron, May 2021.

[17] Greg Beaucage, *Chapter 4 of Polymer Physics: Molecular Motion in Dilute Solution*, University of Cincinnati, April 18, 1998.

[18] Ulrich H. Kurzweg, *Derivation of the Stokes Drag Formula*, University of Florida, November 2003.

[19] Nikola Sandrić, *Stability of the Overdamped Langevin Equation in Double-Well Potential*, Journal of Mathematical Analysis and Applications (2018), **467** (1), 734-750 [DOI: 10.1016/j.jmaa.2018.07.043].

[20] Doyle & Underhill, *Brownian Dynamics Simulations of Polymers and Soft Matter*, in: Handbook of Materials Modeling, edited by S. Yip, 2619-2630, Springer, 2005, [DOI: 10.1007/978-1-4020-3286-8_140].

[21] B. Bird, C. F. Curtiss, R. C. Armstrong, and O. Hassager, *Dynamics of Polymeric Liquids, Volume 2: Kinetic Theory, 2nd Edition*, John Wiley and Sons, 1987.

[22] Michael Cross, *Physics 127a: Class Notes; Lecture 8: Polymers*, California Institute of Technology, October 2006.

[23] ScienceDirect, *Radius of Gyration*, retrieved on 05/01/2021 from `https://www.sciencedirect.com/topics/engineering/radius-of-gyration`.

[24] Huang, Bhattacharya, & Binder, *Conformations, Transverse Fluctuations, and Crossover Dynamics of a Semi–Flexible Chain in Two Dimensions*, The Journal of Chemical Physics (2014) **140** (21), 214902, [DOI: 10.1063/1.4879537].

[25] Soh, Narsimhan, Klotz, Doyle, *Knots Modify the Coil–Stretch Transition in Linear DNA Polymers*, Soft Matter Journal (2018) **14** (9), 5265-5273, [DOI: 10.1039/c7sm02195j].

[26] Rauscher, Rowan, & de Pablo, *Hydrodynamic Interactions in Topologically Linked Ring Polymers*, Physical Review E (2020) **102** (3), 032502, American Physical Society, [DOI: 10.1103/PhysRevE.102.032502].

[27] Tree, Muralidhar, Doyle, & Dorfman, *Is DNA a Good Model Polymer?*, Macromolecules (2013) **46** (20), 8369-8382, [DOI: 10.1021/ma401507f].

[28] Chemistry LibreTexts, *Worm-like Chain*, retreived on 11/05/2022 from `https://chem.libretexts.org/Bookshelves/Biological_Chemistry/Concepts_in_Biophysical_Chemistry_(Tokmakoff)/02%3A_Macromolecules/09%3A_Macromolecular_Mechanics/9.02%3A_Worm-like_Chain`.

[29] PolymerDataBase.com, *Diffusion of Polymer Chains*, retreived on 07/16/2022 from `https://polymerdatabase.com/polymer%20physics/Diffusion.html`.

[30] Diffusion.com, *Diffusion Coefficient*, retrieved on 11/15/2021 from `https://study.com/academy/lesson/diffusion-coefficient-definition-equation-units.html`.

[31] National Institute of Standards and Technology, *Autocorrelation*, retrieved on 08/11/2021 from `https://www.itl.nist.gov/div898/handbook/eda/section3/eda35c.htm`.

Appendix A: Diffusion

Diffusion is the migration of substance from a relatively high concentrated area to areas of lower concentration. Diffusion is related to the concept of Brownian motion as it can be the demonstration of particles' random movements on a macroscopic scale. According to the polymer database: "A small particle diffuses in solution and melt due to fluctuations of the number of nearby molecules randomly colliding with the particle from different directions. Polymer molecules are typically much larger than solvent molecules (or monomers) but still small enough that random collisions with other molecules noticeably move the polymer." An example is given in Figure A.1.

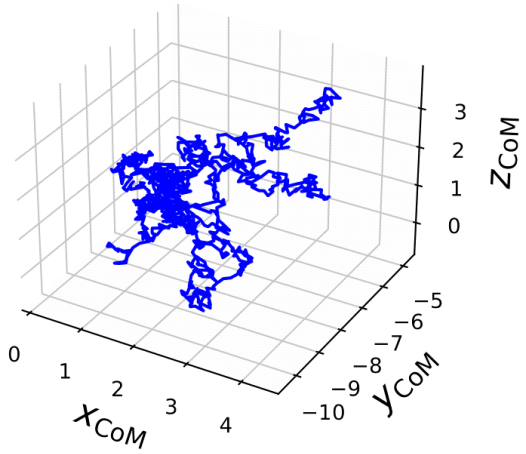


Figure A.1: A sample plot of the diffusion of a 3D Brownian system. The coordinates of the center of mass (CoM) of the system have been plotted over time. Initially, the CoM of the ensemble starts at the origin, but with the progression of time, applied forces like the Brownian forces cause the ensemble to migrate to less concentrated areas.

The *diffusion coefficient* (or *constant*), $\mathcal{D}$, is the rate of diffusion in a stated time. As an instance, how quickly the Brownian system in Figure A.1 spreads from $(x_{\text{CoM}}, y_{\text{CoM}}, z_{\text{CoM}}) = (0,0,0)$ to other points in space determines its diffusion coefficient [30].

Appendix B: Autocorrelation

B.1 The Definition

Autocorrelation is a correlation coefficient between two quantities of the same variable at two separate points in time. With measurements $Y_1, Y_2, \ldots, Y_N$ at times $t_1, t_2, \ldots, t_N$, the lag κ autocorrelation mathematical expression is [31]:

$$\text{AC}_\kappa = \frac{\sum_{j=1}^{N-\kappa}(Y_j - \langle Y\rangle)(Y_{j+\kappa} - \langle Y\rangle)}{\sum_{j=1}^{N}(Y_j - \langle Y\rangle)^2} \tag{B.1}$$

B.2 The Function

The PYTHON 3 function we use to compute our autocorrelation coefficient:

```
def AC(Y, lags):
    average = numpy.average(Y)
    variance = numpy.variance(Y)
    Ys = Y - average
    correlation = numpy.correlate(Ys, Ys, 'full')[len(Y)-1:]/variance/len(Y)
    return correlation[:len(lags)]
```

This function was used to find the relaxation times for the linear real chain and [2]catenane in Section 2.3 and Section 3.3.2 respectively.